ROYAUME DU MAROC
--*-*-*
HAUT COMMISSARIAT AU PLAN
--*-*-*-*-*-*
INSTITUT NATIONAL
DE STATISTIQUE ET D'ECONOMIE APPLIQUEE
INSEA

Projet de Fin d'Etudes

*****

Désambiguïsation des tweets émis lors des catastrophes naturelles

Préparé par : ***M. MOUHASSINE Badreddine***

Sous la direction de :

M. SAIDI Mohamed Nabil **(INSEA)**
Mme. GRACIANNE Cécile **(BRGM)**
Mme. MONTARNAL Aurélie **(IMT Mines Albi)**

Soutenu publiquement comme exigence partielle en vue de l'obtention du

Diplôme d'Ingénieur d'Etat

Filière : ***Data & Software Engineering***

Devant le jury composé de :

- ***M. SAIDI Mohammed Nabil*** (INSEA)
- ***M. BENELALLAM Imade*** (INSEA)
- ***Mme. GRACIANNE Cécile*** (BRGM)
- ***Mme. MONTARNAL Aurélie*** (IMT Mines Albi)

Juin 2023/ PFE N° 51

Dédicace

“

À mes chers parents,

Vous êtes mes piliers, ma source de soutien inébranlable tout au long de ce parcours académique. Votre amour, votre encouragement et votre confiance ont été les forces motrices qui m'ont poussé à atteindre ce jour tant attendu. Votre présence constante dans ma vie est un cadeau précieux que je chérirai toujours,

À mes amis et proches,

votre soutien inconditionnel m'ont porté à travers les hauts et les bas de ce parcours académique.Je tiens à vous remercier du fond du cœur pour votre soutien inébranlable et les souvenirs précieux que nous avons partagés tout au long de cette aventure. Ce travail vous est dédié en témoignage de mon amour et de ma gratitude éternels.

À mes petits frères et soeurs, Qui me sont trop chers et qui savait toujours comment procurer la joie à ma vie.

À tous ceux qui ont joué un rôle dans ma vie, Merci du fond du cœur pour votre amour et votre impact sur ma vie est inestimable . Que cette dédicace témoigne de mon affection et de ma reconnaissance éternelles.

Avec tout mon amour.

”

- Badreddine

Remerciements

D'abord, je remercie Dieu, le Tout-Puissant, pour m'avoir doté du courage et de la patience nécessaires pour mener ce travail à terme.

Je tiens à exprimer toute ma gratitude à mes encadrantes, **Mme. Gracianne Cécile** et **Mme. Montarnal Aurélie**. Leur soutien expert, leur patience et leurs encouragements ont été fondamentaux dans ce projet. Leurs conseils judicieux ont été précieux pour structurer mon travail et pour améliorer la qualité des différentes sections. Leur aide inestimable et la qualité de leur suivi m'ont été d'une grande valeur, tout comme les conseils et les informations qu'elles m'ont prodigués avec une remarquable patience et un professionnalisme sans faille.

Je souhaite adresser un très grand remerciement et une reconnaissance profonde à **M. Mohammed Nabil Saidi**, mon encadrant à l'INSEA. Ses précieuses connaissances en matière d'apprentissage automatique, ainsi que son encadrement lors de mon stage, ses conseils et son soutien tout au long de mon parcours au cycle d'ingénieur à l'INSEA, ont été indispensables. J'apprécie également ses écoutes et conseils apportés au fil de différents projets parascolaires. Un merci tout particulier à **Mme. Saidi Rajaa**, notre professeure et ancienne coordinatrice de la filière DSE, pour tous ses efforts et son soutien qui nous ont aidé à représenter l'INSEA et la filière lors de divers événements nationaux, et pour sa contribution à l'amélioration de notre formation.

Je tiens à remercier le chef du projet RéSoCIO, **M. Auclair Samuel**, pour son soutien et l'opportunité qu'il m'a donnée de travailler sur le projet RéSoCIO et de rencontrer plusieurs collaborateurs.

Ma reconnaissance s'étend également à l'assistante RH, **Mme. Pezard Isabelle**, pour son aide et son soutien administratif, sa disponibilité et ses orientations durant la phase d'organisation de ma mobilité. Ses efforts dans tous les processus précédant mon arrivée et mon début de stage au BRGM ont été inestimables.

Je remercie toute l'équipe de l'unité DRP/RSV pour le merveilleux environnement de travail, leur sympathie, leur aide précieuse, leurs encouragements et pour avoir rendu mon stage une expérience très enrichissante.

À tous les membres du jury, veuillez recevoir mes sincères remerciements pour l'honneur que vous me faites en prenant le temps de lire et d'évaluer ce travail. Je souhaite également exprimer ma gratitude à tous les profs de la filière DSE et à l'équipe pédagogique et administrative de l'INSEA pour leurs efforts afin de nous offrir une excellente formation.

Enfin, je souhaite remercier toutes les personnes ayant contribué, de près ou de loin, à la réalisation de ce travail.

Résumé

Ce travail de fin d'études a pour objectif d'améliorer l'extraction d'informations relatives aux situations de crise à partir de données provenant de Twitter. Il consiste à regrouper les tweets traitant un même événement, ce qui permet d'optimiser la gestion des crises. Ce projet s'inscrit dans le cadre du Projet National de Recherche RéSoCIO, dont l'objectif est de créer un système de gestion des catastrophes naturelles en utilisant les données des réseaux sociaux.

Dans un premier temps, nous avons analysé différentes méthodes existantes dans l'état de l'art et la méthode SCCL basé sur le *Short-text clustering* semble être une solution appropriée pour notre tâche. Nous avons utilisé plusieurs bases de données contenant des tweets en anglais et en français pour effectuer des tests et des évaluations. Les résultats sont prometteurs, notamment pour les bases de données en anglais. Cependant, nous avons rencontré des difficultés lors du traitement des données en français.

Par ailleurs, nous avons exploré une autre approche basée sur l'apprentissage de la similarité. Nous avons utilisé des techniques de transfert d'apprentissage et l'apprentissage contrastif pour améliorer l'encodeur de tweets en français en utilisant des adaptateurs de *Transformers* et un seuil de similarité dynamique pour catégoriser les paires de tweets traitant du même événement. Nous avons testé différentes configurations et paramètres pour cette méthode, et les configurations de Houlsby ont donné de bons résultats sur les bases de données de test et de validation. Cependant, des améliorations supplémentaires sont encore nécessaires, notamment lors du traitement de bases de données déséquilibrées.

Dans l'ensemble, ce projet contribue à exploiter les réseaux sociaux pour la gestion des crises en facilitant l'extraction d'informations pertinentes à partir de Twitter. Bien qu'il reste des améliorations à apporter, ce travail représente une première étape importante pour les travaux futurs dans ce domaine.

Mots clés : tweets, réseaux sociaux, gestion des crises, *Transformers*, *Short text clustering*, apprentissage par transrfert, adaptateurs des *Transformers*, apprentissage contrastif, apprentissage de similarité.

Abstract

This final year project focuses on improving the retrieval of crisis-related information from Twitter data by identifying and grouping tweets that discuss the same event, which leads to more effective crisis management. The project is part of the National Research Project RéSoCIO, which aims to develop a system for managing natural disasters using social media data.

Firstly, we explored various state-of-the-art methods. The SCCL method emerged as a suitable solution for the task. We utilized various datasets, including both English and French tweets, for experimentation and evaluation. The results showed promising potential, particularly with English datasets. However, we encountered problems when dealing with French datasets.

Additionally, we explored an alternative approach based on similarity learning. This involved utilizing transfer learning and contrastive learning techniques to fine-tune the French tweet encoder. We used transformer adapters and a dynamic similarity threshold to classify pairs of tweets discussing the same event. We experimented with various configurations and parameters for this approach. The Houlsby configuration showed good performance in both test and validation datasets, but requires further refinement, particularly when dealing with imbalanced datasets.

Overall, this project contributes to leveraging social media for crisis management, facilitating the retrieval of valuable information from Twitter. While further improvements are needed, this work lays a solid foundation for ongoing studies in this subject area.

Keywords : tweets, social media, crisis management,Transformers, short-text clustering, , transfer learning, transformer adapters, contrastive learning, similarity learning.

Table des matières

Dédicace . . . . . **I**

Remerciements . . . . . **II**

Résumé . . . . . **III**

Abstract . . . . . **IV**

Introduction générale **1**

I Contexte général et organisme d'accueil **5**
1 Le Bureau de Recherches Géologiques et Minières 6
2 L'Ecole des Mines-télécom "*IMT Mines Albi* " 8
3 Contexte du travail 8
3.1 Le projet RéSoCIO 8
3.2 Structure du projet 9
4 Organisation du stage 11
4.1 Planification du projet de stage 11
4.2 Environnement et ressources techniques 12
5 Synthèse 13

II Revue de la littérature **15**
1 Le traitement automatique du langage naturel (TALN) 16
1.1 Histoire du TALN 16
1.2 Applications du TALN 18
1.3 Enjeux du TALN 19
1.4 Conclusion 20
2 les *Transformers* 20
2.1 Le mécanisme d'Attention 20
2.2 L'architecture des *Transformers* 21
2.3 Les adaptateurs pour les *Transformers* 23
2.4 Les *Transformers* pré-entrainé 24
2.5 Conclusion 26
3 Etat de l'art 26
3.1 La ségmentation du texte 26
3.2 Approches basées sur les graphes 27
3.3 Approches basées sur la ségmentation classique 29
4 Synthèse 30

III Présentation des bases de données . . . 31
1 Introduction au Twitter . . . 32
1.1 L'API du Twitter . . . 32
1.2 Caractéristiques générales des tweets . . . 33
1.3 Préparation des données . . . 34
1.4 Conclusion . . . 34
2 Données CrisisNLP HumAID . . . 34
3 Données RéSoCIO . . . 37
4 Données filtrées . . . 41
5 Synthèse . . . 43

IV Approche basée sur le clustering . . . 44
1 Introduction . . . 45
1.1 L'apprentissage contrastif . . . 45
1.2 L'augmentation des données textuelles . . . 46
2 Architecture du modèle . . . 48
3 Expérimentation et résultats . . . 50
3.1 Expérimentation 1 . . . 51
3.2 Expérimentation 2 . . . 54
3.3 Expérimentation 3 . . . 56
4 Synthèse . . . 58

V Approche basé sur l'apprentissage de similarité . . . 59
1 Introduction . . . 60
1.1 Mesure de la similarité . . . 60
1.2 Préparation des données . . . 61
2 Architecture du modèle . . . 62
3 Expérimentations . . . 63
3.1 Expérimentation avec les adaptateurs de type "Bottleneck Adapters" 63
3.2 Expérimentation avec l'adaptateur de type "Compacter" . . . 67
3.3 Test final . . . 69
4 Synthèse . . . 71

Conclusion et perspectives . . . 73

Bibliographie . . . 75

Annexes . . . 77

A Technologies . . . 78

B Formules et Algorithmes . . . 80

Table des figures

1.1 Logo du BRGM 6
1.2 Organigramme générale du BRGM 7
1.3 Logo du projet RéSoCIO 9
1.4 Découpage du projet RéSoCIO 10
1.5 Diagramme de GANTT pour l'organisation du projet de stage 11

2.1 L'architecture du Transformer [21] 22
2.2 Exemple d'un adaptateur *Transformer* 23
2.3 Procédures globales d'entrainement et de fine tuning pour BERT 24
2.4 l'architecture CLARE 28

3.1 Logo du Twitter 32
3.2 Nombre des tweets par événement sur les données HumanAID 35
3.3 les mots très fréquent dans les tweets sur le séismes de Kaikoura (Word Cloud) 36
3.4 les mots très fréquent dans les tweets sur le séismes de l'Ecuador (Word Cloud) 37
3.5 les mots très fréquent dans les tweets sur le séismes de l'italie (Word Cloud) 37
3.6 Nombre des tweets par événement de type séismes sur les données RéSoCIO 38
3.7 Cas de chevauchement des catstrophes naturelles dans le temps 39
3.8 Nombre des tweets par événement de type inondations sur les données RéSoCIO 39
3.9 les mots très fréquent dans les tweets sur les inondations dans les données RéSoCIO 40
3.10 les mots très fréquent dans les tweets sur les séismes dans les données RéSoCIO 41
3.11 Nombre des tweets par événement sur les données filtré 41
3.12 les mots très fréquent dans les tweets sur les événements des données filtré 42

4.1 Démonstration du processus clés de l'apprentissage contrastif 45
4.2 Framework d'entrainement SCCL [23] 48
4.3 Evaluation sur SearchSnipets au niveau des clusters [23] 49
4.4 Évaluation de la convergence de la clustering loss lors de la validation dans l'expérimentation 1 52
4.5 Évaluation de du NMI lors de la validation dans l'expérimentation 1 52
4.6 Évaluation du clustering accuracy lors de la validation dans l'expérimentation 1 53
4.7 Evaluation de la convergence du clustering loss lors de la validation dans l'expérimentation 2 55

4.8 Evaluation de du NMI lors de la validation dans l'expérimentation 2 55
4.9 Evaluation du clustering accuracy lors de la validation dans l'expérimentation 2 55
4.10 Evaluation de la convergence du clustering loss lors de la validation dans l'expérimentation 3 57
4.11 Evaluation de du NMI lors de la validation dans l'expérimentation 3 57
4.12 Evaluation du clustering accuracy lors de la validation dans l'expérimentation 3 58

5.1 Architecture de notre modèle de prédiction des similarités 62
5.2 Architecture des configurations BottleNeck Adapters [16] 64
5.3 la fonction du *contrastive loss* pendant l'entrainement du modèle avec la configuration houlsby 65
5.4 l'accuracy de validation pendant l'entrainement du modèle avec la configuration houlsby 65
5.5 la fonction du *contrastive loss* pendant l'entrainement du modèle avec la configuration pfeiffer 66
5.6 l'accuracy de validation pendant l'entrainement du modèle avec la configuration pfeiffer 67
5.7 la fonction du *contrastive loss* pendant l'entrainement du modèle avec la configuration Compacter 68
5.8 l'accuracy de validation pendant l'entrainement du modèle avec la configuration Compacter 68
5.9 matrice de confusion du test final du modèle 69

Liste des tableaux

1.1 Librairie et version utilisée dans le projet du stage 13
1.2 Configuration de lancement des expérimentation sur le Leto 13

4.1 Configuration du SCCL de la première Expérimentation 51
4.2 Configuration du SCCL de la deuxième expérimentation 54
4.3 Configuration du SCCL de la troisième expérimentation 56

5.1 Configuration d'entraînement du modèle 63
5.2 métriques de classification des similarités 70

Liste des sigles et acronymes

API	*Application Programming Interface*
BERT	*Bidirectional Encoder Representations from Transformers*
BRGM	*Bureau de Recherches Géologiques et Minières*
CATNAT	*Catastrophes Naturelles*
CGI	*Centre Génie Industrielle*
DRP/RSV	*Direction Risques et prévention/ Risques Sismiques et Volcaniques*
IMT	*Institu Mines Télécom*
Instane-CL	*Instance Contrastive Learning*
NMI	*Normalized Mutual Information*
RS	*Réseau Sociaux*
SCCL	*Supporting sclustering with contrastive learning*
TALN	*Traitement Automatique de Langage Naturelle*

Introduction générale

Contexte

À l'ère actuelle du numérique, les réseaux sociaux ont acquis une importance considérable en devenant des vecteurs primordiaux de communication et d'échange d'informations. Ces plateformes dynamiques permettent aux utilisateurs du monde entier de partager et de recevoir une multitude d'informations en un temps record, révolutionnant ainsi notre manière de communiquer.

Les réseaux sociaux offrent une palette variée de possibilités. Parmi ces dernières, la collecte d'informations et l'accès à l'actualité en temps réel se révèlent particulièrement précieuses. Que ce soit pour suivre le déroulement d'un événement sportif, politique ou encore culturel, ces plateformes numériques permettent de rester constamment informé. De plus, elles ont la capacité de fournir divers types d'informations, allant des plus simples aux plus complexes, ce qui facilite grandement l'accès à l'information pour tous.

En outre, les réseaux sociaux s'avèrent d'une grande utilité lors de situations d'urgence. En cas de catastrophe naturelle, par exemple, ils peuvent servir de canal de communication rapide et efficace pour diffuser des informations cruciales, des consignes de sécurité, ou encore pour organiser des actions d'aide et de secours. Ils permettent ainsi d'obtenir des mises à jour en temps réel sur la situation, ce qui peut s'avérer déterminant pour gérer l'événement de la meilleure manière possible.

Dans le cadre de notre projet, la capacité des réseaux sociaux à fournir des informations en temps réel et à grande échelle est une ressource précieuse. En effet, cela nous permet d'avoir une compréhension claire et rapide de la situation, et ainsi de prendre des décisions informées et appropriées. La richesse et la rapidité de ces données contribuent à nous aider à mieux naviguer dans un monde en constante évolution et à répondre efficacement aux défis qui se présentent.

Le BRGM collabore avec l'IMT Mines Albi et d'autres partenaires de recherche scientifique dans le cadre du projet RéSoCIO, qui vise à développer un système pour gérer les catastrophes naturelles en utilisant les données des réseaux sociaux. Notre stage s'inscrit dans ce projet dans un cadre de recherche scientifique, où nous cherchons à améliorer l'obtention des informations générales sur une crise à partir des tweets, qui sont des messages textuels publiés sur le réseau social Twitter.

Problématique

Les réseaux sociaux comme Twitter sont des sources d'informations précieuses, surtout durant les catastrophes naturelles. Leur rapidité et le grand nombre de messages échangés nous permettent d'obtenir des informations en temps réel. Pour l'instant, quand on observe une série de tweets, on analyse habituellement chaque tweet un par un. Cela signifie que pour comprendre un événement dans sa globalité, on doit passer en revue tous les tweets liés à cet événement. Le souci est qu'il n'est pas toujours certain que tous ces tweets se réfèrent au même événement, en particulier s'ils sont publiés en peu de temps.

À ce jour, le BRGM peut rassembler des tweets relatifs à certaines crises, comme les tremblements de terre ou les inondations. Mais ceux qui gèrent ces crises ont besoin d'informations plus précises, comme le lieu exact ou des détails sur les victimes. Ces informations sont essentielles pour organiser les secours et gérer la crise efficacement.

Néanmoins, si plusieurs crises surviennent simultanément, par exemple deux tremblements de terre à différents endroits, on risque de se tromper en pensant qu'il n'y en a qu'un seul. C'est un problème car cela peut retarder l'aide nécessaire. Notre principal défi est donc de trouver comment repérer les tweets qui parlent du même événement. Cela nous aiderait à comprendre plus clairement la situation et à fournir les bonnes informations à ceux qui gèrent la crise.

En résumé, notre principal problème est de trouver un moyen d'identifier et de regrouper les tweets se rapportant au même événement. En résolvant ce problème, nous pourrions rendre la surveillance des réseaux sociaux plus efficace et donner aux décideurs les informations précises dont ils ont besoin en cas de crise. Cependant, cela pose une question importante : **comment identifier les tweets qui discutent du même événement ?**

Objectifs

Dans le cadre de notre stage, notre objectif principal est d'améliorer la compréhension globale de la crise à l'aide de l'information contenue dans les tweets à un instant donnée. Étant donné que nous traitons des données textuelles provenant des réseaux sociaux, l'une de nos premières étapes consiste à analyser et à obtenir une vue d'ensemble des données que nous allons tester. Ensuite, nous appliquons des techniques d'intelligence artificielle qui permettent de traiter efficacement les données textuelles, y compris des méthodes basées sur l'apprentissage profond.

Nous cherchons à créer un modèle capable de distinguer les données représentant différentes catastrophes naturelles. Pour cela, nous utilisons des approches avancées de l'intelligence artificielle afin de reconnaître les schémas et les caractéristiques spécifiques associées à chaque type de catastrophe naturelle.

En résumé, il s'agira de désambiguïser les tweets concernant les crises naturelles afin de pouvoir associer l'ensemble des informations complémentaires qui seraient présentes dans

plusieurs tweets distincts.Nous analysons d'abord les données, puis nous appliquons des techniques, notamment l'apprentissage profond, pour entraîner notre modèle à distinguer les différentes données représentant divers types de catastrophes naturelles.

Organisation du rapport

Ce rapport est organisé en cinq chapitres :
Le premier chapitre intitulé "**Organisme d'accueil**" a pour objectif de présenter l'organisme qui nous accueille pendant la durée de notre stage dans un cadre scientifique de recherche et développement. Nous mettrons en évidence ses missions principales ainsi que l'unité au sein de laquelle nous réalisons notre travail. Ensuite, nous aborderons le projet dans lequel notre sujet de stage s'inscrit. Nous présenterons la structure globale du projet et préciserons dans quelle partie spécifique du projet nous allons travailler.

Le deuxième chapitre, intitulé "**État de l'art**", se concentre sur notre étude bibliographique concernant le traitement automatique du langage naturel en général, ainsi que les techniques utilisées pour regrouper et distinguer les données textuelles. Dans ce chapitre, nous présenterons quelques définitions des termes clés utilisés dans la littérature. Ensuite, nous mettrons en évidence les modèles et les publications les plus récents et les tendances actuelles dans le domaine du traitement du texte, en particulier grâce aux techniques d'apprentissage profond.

Le troisième chapitre intitulé "**Présentation des données**", contient une analyse générale des données que nous utiliserons ultérieurement pour l'entraînement et l'évaluation de nos modèles. Tout d'abord, nous présenterons le pipeline de préparation de ces données, en expliquant leur source, leur nature et les différentes caractéristiques spécifiques de ces types de données. Ensuite, nous procéderons à une brève analyse exploratoire de ces données, afin de mieux les comprendre et d'identifier des tendances ou des informations pertinentes.

Le quatrième chapitre "**Solution basée sur le clustering**", nous allons vous parler du premier travail que nous avons accompli après avoir étudié les travaux déjà existants.Le chapitre débutera par une introduction où nous expliquerons les techniques et les principes importants pour bien comprendre l'architecture du modèle. Ensuite, nous vous présenterons l'architecture du modèle, et enfin, nous vous parlerons des expérimentations que nous avons menées dans ce domaine.

Le cinquième chapitre "**Solution basé sur l'apprentissage du similarité**", nous aborderons le deuxième travail effectué après avoir expérimenté et analysé les résultats du travail précédent. Nous avons choisi d'utiliser l'apprentissage de similarité, en utilisant le *Transfer Learning*, afin de construire un modèle capable de prédire si deux tweets traitent de la même crise ou non. Ainsi, dans ce chapitre, nous vous présenterons en détail l'architecture du modèle et explorerons les différentes expérimentations que nous mènerons avec ce modèle.

À la fin du rapport, nous conclurons par une conclusion générale qui récapitulera tout le travail réalisé pendant le stage, mettra en avant les résultats obtenus, les contributions apportées, et soulignera les principales conclusions ainsi que les points forts et les limitations du travail accompli. Nous discuterons également des perspectives d'évolution, des possibilités d'amélioration et des extensions éventuelles du projet.

Chapitre I

Contexte général et organisme d'accueil

Dans ce chapitre, nous présenterons l'entreprise qui nous a accueilli pendant notre stage, en mettant en avant sa mission principale ainsi que l'unité dans laquelle notre travail s'est déroulé. De plus, nous explorerons le projet global qui constitue le contexte de notre stage, en décrivant ses différents composants et en mettant en évidence les différents lots du projet, dont celui dans lequel notre travail s'inscrit. Ce chapitre sera une présentation essentielle pour comprendre le contexte et l'environnement dans lesquels nous avons évolué pendant cette expérience professionnelle.

1 Le Bureau de Recherches Géologiques et Minières

Fig. 1.1 : Logo du BRGM

Le B.R.G.M (Bureau de Recherches Géologiques et Minières) est le service géologique national français qui a pour mission de gérer les ressources et les risques du sol et du sous-sol en utilisant les sciences de la Terre. En tant qu'un établissement public à caractère industriel et commercial crée en 1959, le BRGM s'engage auprès de l'état à travers son contrat d'objectifs de moyen et de performance (COMP) qui définit les principales actions qu'il doit conduire au cours de la période 2023-2027. Le BRGM est partenaire de nombreux acteurs publics et privés et met la recherche scientifique, l'expertise et l'innovation au cœur de ses actions pour répondre à quatre objectifs principaux : comprendre les phénomènes géologiques et les risques associés, développer des méthodologies et des techniques nouvelles, produire et diffuser des données pour la gestion du sol, du sous-sol et des ressources, et mettre à disposition les outils nécessaires à la gestion du sol, du sous-sol et des ressources, à la prévention des risques et des pollutions, aux politiques de réponse au changement climatique. L'action du BRGM s'inscrit dans le cadre de six grands enjeux scientifiques et sociaux :

- la géologie et la connaissance du sous-sol
- la gestion des eaux souterraines
- les risques et l'aménagement du territoire
- les ressources minérales et l'économie circulaire
- la transition énergétique et l'espace souterrain
- les données, services et infrastructures numériques

Pour répondre à ces enjeux, le BRGM a établi une stratégie scientifique pour une meilleure lisibilité de ses actions. Cette stratégie, mise à jour en 2019, définit les priorités de recherche du service géologique national pour les dix prochaines années et se structure en missions transverses et grands secteurs d'applications . Grâce à ses actions, le BRGM

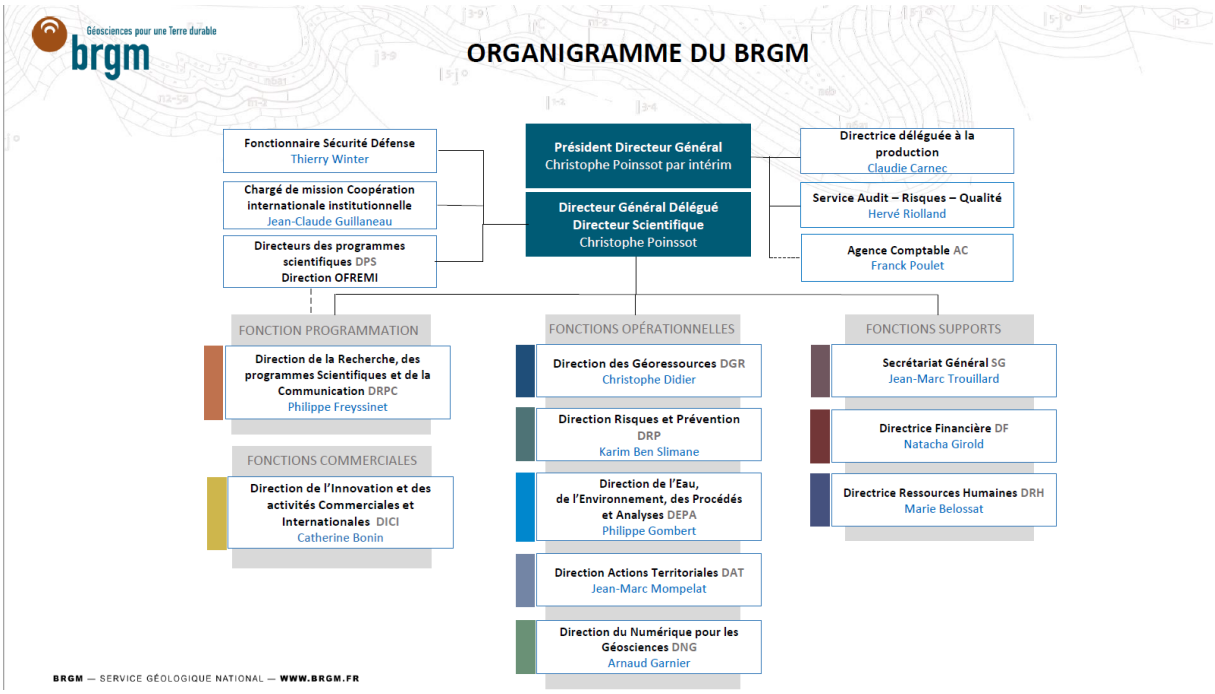

Fig. 1.2 : Organigramme générale du BRGM

contribue à répondre aux grands défis de notre société, tels que le changement climatique, la transition énergétique et le développement de l'économie circulaire.

Le BRGM se compose de plusieurs directions(Figure 1.2), parmi ces directions, on trouve la direction où se déroule notre stage, La Direction des Risques et de la Prévention (DRP) qui est responsable d'un large éventail de compétences dans différents domaines y compris :

- Les risques naturels, en traitant des séismes, des mouvements de terrain, des tsunamis, du volcanisme, de l'érosion des sols, de l'élévation du niveau de la mer, de l'érosion du littoral, des submersions marines, et bien d'autres encore.
- L'évaluation de l'impact du changement global, tels que le changement climatique, ainsi que l'évolution de l'exposition aux aléas.
- Les risques liés aux cavités souterraines, à l'après-mine, aux stockages souterrains tels que le CO2 et l'énergie, ainsi qu'aux exploitations du sous-sol font également partie de ses compétences.

L'unité DRP/RSV (Risques Sismiques et Volcaniques) Notre stage principalement se déroule au sein de l'unité RSV qui a pour mission de développer, entretenir et partager la connaissance des risques sismiques et volcaniques, ainsi que d'apporter un soutien aux autorités publiques et aux acteurs privés dans la gestion de ces risques. Leur travail englobe une gamme étendue de compétences scientifiques et techniques dans l'analyse du risque sismique, allant de l'étude de la source sismique à l'évaluation de la réponse des sols, en passant par l'évaluation de la vulnérabilité physique ou fonctionnelle des structures.

Dans le cadre de la gestion des risques via les réseaux sociaux, le BRGM a développé une plateforme appelée SURICATE-Nat. Cette plateforme analyse automatiquement les tweets liés aux catastrophes naturelles. Son objectif est de permettre aux "citoyens capteurs" de partager rapidement des informations importantes. Actuellement, la plateforme collecte des tweets concernant différents types de catastrophes naturelles, les localise au niveau de la commune et extrait des informations pertinentes. Elle est en cours de révision afin d'améliorer sa capacité à traiter un grand nombre de tweets en cas de forte affluence. Son architecture est conçue pour gérer les événements et allouer dynamiquement de nouvelles ressources en fonction du nombre de tweets à analyser.

2 L'Ecole des Mines-télécom "*IMT Mines Albi*"

L'IMT Mines Albi est une institution d'enseignement supérieur située à Albi, en France. Reconnue pour son engagement dans la recherche scientifique, cette école vise à repousser les frontières de la connaissance. Elle aborde des domaines tels que l'ingénierie, les matériaux avancés et les technologies énergétiques, dans le but de résoudre des problèmes concrets et d'apporter des solutions innovantes pour améliorer notre quotidien. Grâce à une équipe de chercheurs passionnés, IMT Mines Albi se positionne comme un moteur de progrès scientifique et technologique dans la région.

Au sein d'IMT Mines Albi, le Centre Génie Industriel (CGI) se distingue par son expertise rare et sa combinaison unique de compétences en Génie Industriel et en Intelligence Artificielle. Fondé en 1996, le CGI se focalise sur des domaines tels que la science des données, l'ingénierie des modèles, l'ingénierie à base de connaissances et la recherche opérationnelle. En étroite collaboration avec divers secteurs, le CGI s'efforce de fournir aux entreprises des solutions opérationnelles de haute qualité. Ses recherches couvrent une vaste gamme de domaines, allant des services de transport à la rénovation énergétique, en passant par les crises humanitaires et les soins à domicile. De plus, les approches développées dans le domaine manufacturier peuvent être adaptées à d'autres contextes. Le CGI exploite les nouvelles technologies de l'information et de la communication pour relever les défis posés par des environnements en constante évolution, caractérisés par des collaborations multipartites, des changements fréquents, des incertitudes importantes, une diversité et une complexité croissantes.

Dans le cadre de notre stage, nous travaillons en collaboration avec IMT Mines Albi, qui mène depuis longtemps des recherches dans le CGI sur la gestion des crises en général, et en particulier sur le projet RéSoCIO, en collaboration avec le BRGM et d'autres partenaires.

3 Contexte du travail

3.1 Le projet RéSoCIO

Le projet RéSoCIO est un projet de recherche visant à explorer les possibilités offertes par les réseaux sociaux (RS) pour aider à la gestion opérationnelle des catastrophes naturelles (CATNAT). L'objectif principal du projet est de démontrer comment les RS

Fig. 1.3 : Logo du projet RéSoCIO

peuvent fournir des informations en temps réel depuis le terrain, ce qui peut être particulièrement utile pour les phénomènes à cinétique rapide qui peuvent prendre de court les autorités. En permettant la collecte, la consolidation, la précision et l'augmentation des informations disponibles pour les décideurs, cette approche peut fournir un complément d'informations qui pourrait aider à la prise de décision dans les situations de crise.

En plus, le projet vise à répondre à un besoin important dans la gestion de crise, à savoir le manque d'informations précises et fiables qui peut survenir dans les premiers moments d'une catastrophe naturelle. Les RS peuvent être utilisés pour combler cette lacune en fournissant des informations en temps réel sur les zones touchées, les problèmes de sécurité, les besoins en secours et les ressources disponibles. Les gestionnaires de crise peuvent ainsi prendre des décisions plus éclairées et mieux coordonner les opérations de secours. Il entend proposer des éléments de contexte qui pourraient être utiles aux praticiens de la gestion de crise. Par exemple, dans la situation où les autorités ne disposent pas d'informations sur l'emplacement ou l'ampleur de la catastrophe, les RS pourraient aider à identifier les zones les plus touchées et les ressources nécessaires pour y faire face. En fournissant des informations sur les zones où les interventions sont les plus urgentes et les plus nécessaires, les RS pourraient également aider à éviter le gaspillage des ressources et à optimiser l'efficacité des opérations de secours.

Le projet RéSoCIO est ambitieux et donc il n'est pas sans défis. Les informations provenant des RS doivent être vérifiées pour éviter les fausses alertes et les informations erronées. Les données collectées doivent également être analysées rapidement et efficacement pour être utiles aux gestionnaires de crise. Le projet nécessite également la collaboration de différents acteurs, notamment les praticiens de la gestion de crise, les autorités locales, les organisations de secours et les utilisateurs de RS. Cependant, si ces défis peuvent être relevés, le projet RéSoCIO pourrait contribuer de manière significative à la gestion des catastrophes naturelles.

3.2 Structure du projet

Le projet RéSoCIO, se découpe en cinq lots de travail (voir la figure 1.4) :

Le premier lot LT0 nommée « Gestion du projet et dissémination » est une partie transverse qui vise à piloter le projet RéSoCIO dans sa globalité. Elle est destinée à assurer une bonne exécution du projet en garantissant la cohérence des résultats obtenus avec les objectifs initiaux, les délais et les limites budgétaires envisagés. Cette tâche comprend plusieurs sous-tâches telles que l'animation du projet, la gestion de la cohérence globale, l'interaction avec l'ANR, la dissémination et la valorisation des résultats, ainsi que le management des données. Les objectifs de cette tâche incluent le suivi, le contrôle qualitatif

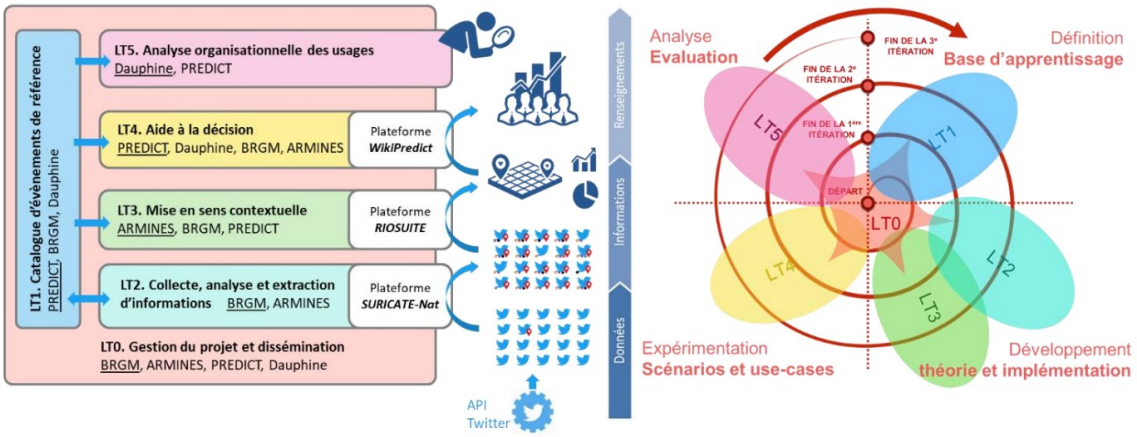

FIG. 1.4 : Découpage du projet RéSoCIO

du projet, la gestion de la planification et des interactions entre les acteurs du projet.

Le LT1 « Catalogue descriptif d'évènements de référence » consiste à fournir aux LT techniques des données communes pour leurs travaux en collectant, synthétisant et organisant les données pour une liste d'événements des catastrophes nationales de référence récente. Pour chaque événement, il s'agit de disposer d'une contextualisation générale, d'une collection de tweets bruts, de données expertes thématiques décrivant l'événement, et d'éléments relatifs à sa gestion opérationnelle. Les tâches comprennent l'élaboration d'une liste d'événements significatifs, la constitution d'un jeu de tweets, la synthèse des données descriptives et l'analyse rétrospective de la gestion opérationnelle.

Le LT2 « Collecte, analyse et extraction d'informations » du projet RéSoCIO vise à assurer la collecte dynamique de tweets pertinents, à analyser leur contenu et leurs métadonnées pour extraire des informations thématiques et géographiques, et à enrichir automatiquement les tweets pour améliorer l'extraction d'informations génériques et spécifiques aux crues éclair et aux séismes. Les tâches incluent l'optimisation des requêtes par mots-clés et l'ajout d'un module de requête spatiale, la caractérisation des collections de tweets, l'enrichissement automatique des tweets pour l'extraction d'informations génériques, et l'enrichissement automatique des tweets pour l'extraction d'informations thématiques.

LT3 vise à regrouper les tweets en clusters afin d'extraire un modèle global de la crise et améliorer la connaissance situationnelle des gestionnaires de crise. La première tâche consiste à définir un cadre multidimensionnel permettant d'associer chaque tweet à un contexte en fonction de caractéristiques présentes dans les données ou méta-données des tweets. Ensuite, une tâche de clustering sera réalisée pour regrouper les tweets parlant des mêmes situations et émergeront des modèles contextuels de situation de crise. Enfin, les clusters identifiés seront comparés à des données exogènes pour identifier leur complémentarité et établir des règles d'analyse conjointe, tels que l'utilisation de notifications pour établir des cartes de probabilité conditionnelle qu'un usager Twitter puisse être le témoin d'un événement.

Le LT4 vise à fournir des informations d'aide à la décision pour les crues éclair ou les séismes à partir des travaux des LT2 et LT3, en se concentrant sur des questions ergonomiques et organisationnelles en collaboration avec le comité utilisateurs. Les spécifications

fonctionnelles incluent l'identification des informations cruciales, des types d'indicateurs et du niveau de qualification requis pour chaque cas d'utilisation. Un exercice final sera réalisé pour évaluer la qualité du projet.

Le LT5 du projet vise à intégrer la dimension organisationnelle dans le développement d'un dispositif de gestion de crise. Le projet comprend une expérience de conception pour analyser les effets de l'utilisation des fonctionnalités RéSoCIO sur le traitement de l'information, ainsi qu'une observation des pratiques et des dynamiques organisationnelles lors d'exercices sur table. De plus, le projet identifiera des axes de vigilance liés à l'utilisation du dispositif, à son adoption et à la transformation organisationnelle associée. L'analyse fournira des informations sur l'adoption du dispositif et les dynamiques organisationnelles générées par sa mise en œuvre.

Les développements réalisés dans le cadre de notre stage de fin d'étude s'adossent aux objectifs du LT2 et du LT3, qui visent à créer un modèle permettant de mettre en contexte et regrouper les tweets émis lors d'un même événement, dans le but de généraliser la connaissance situationnelle des crises et favoriser la prise de décision lors de crues éclair ou de séismes.

4 Organisation du stage

4.1 Planification du projet de stage

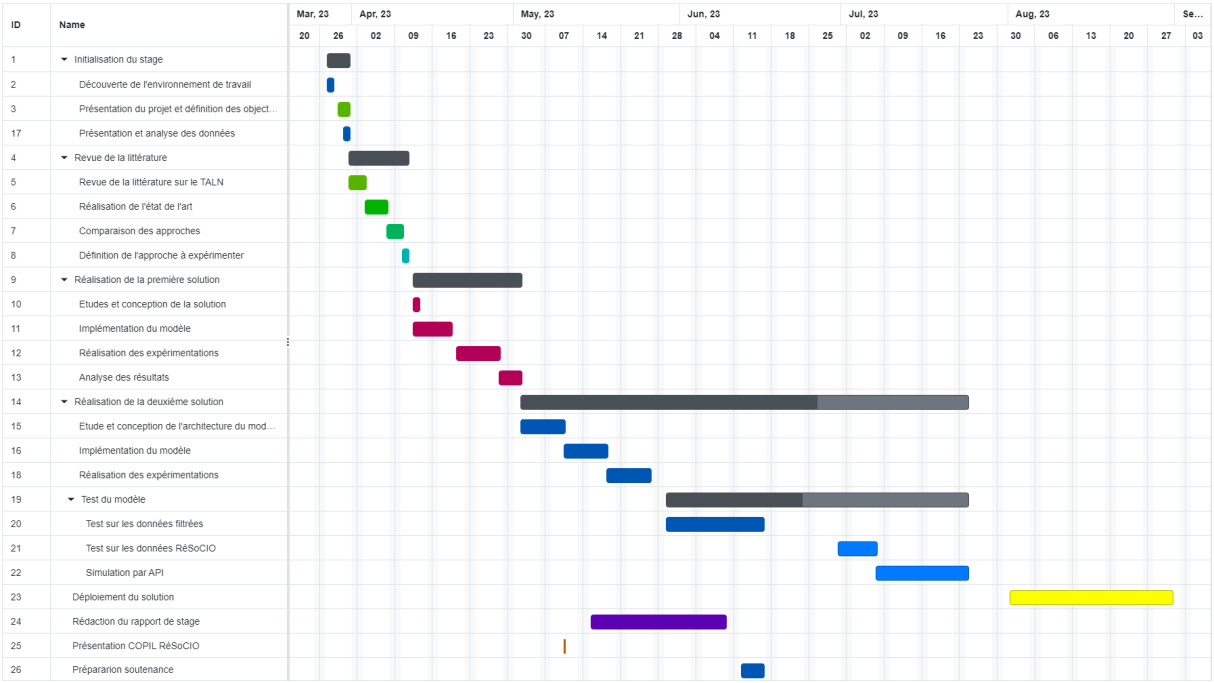

FIG. 1.5 : Diagramme de GANTT pour l'organisation du projet de stage

Notre stage sera organisé en 6 tâches principales, chacune étant composée de plusieurs sous-tâches. Le stage se déroulera au BRGM avec une supervision du BRGM et de l'IMT Mines Albi. Afin de suivre correctement l'avancement du projet, nous avons planifié deux

réunions par semaine, les mercredis et vendredis, pour faire le point sur les travaux et les différentes tâches. Voici les tâches telles qu'elles sont présentées dans le diagramme de Gantt (voir figure 1.5) :

- Initialisation du stage : il s'agit de la phase de démarrage où nous nous familiariserons avec l'environnement de travail, les concepts métier liés au sujet, ainsi que la définition des objectifs et l'exploration des données.
- Revue de la littérature : cette tâche a pour but d'explorer et d'enrichir nos connaissances techniques et théoriques sur le traitement automatique du langage naturel en général, ainsi que sur les modèles existants concernant notre problématique spécifique.
- Réalisation de la première solution : ici, l'objectif principal est d'expérimenter et de tester le modèle pour notre cas et notre problématique. Dans un premier temps, nous effectuerons une étude et une conception des différents composants de l'architecture du modèle, puis nous les implémenterons afin de les expérimenter sur les différentes bases de données dont nous disposons, pour en tirer des conclusions.
- Réalisation de la deuxième solution : cette tâche est similaire à la précédente, mais elle est un peu plus longue car nous devons effectuer une étude, une conception et une implémentation sans disposer d'un modèle de base. C'est la tâche qui prendra le plus de temps, car nous devrons mettre en place tous les composants nécessaires pour notre modèle et réaliser différentes expérimentations afin d'analyser en détail et tirer des conclusions sur ce modèle.
- Test du modèle : cette tâche est la plus importante car elle consiste à effectuer différents tests pour valider le modèle avant de passer à sa production. Nous réaliserons également une simulation en temps réel pour évaluer le comportement de la solution sur différentes données.
- Déploiement de la solution : cette phase finale vise à déployer notre solution en tant que produit et à interagir avec les différents composants du projet RéSoCIO.

4.2 Environnement et ressources techniques

le cluster régional *Leto* est un serveur de calcul utilisé par le BRGM et d'autres collaborateurs dans la région Centre-Val de Loire en France. Il est composé de différents types de nœuds. Les nœuds minces sont équipés de 24 processeurs, 64 Go de RAM et ont 24 cœurs de calcul. Les nœuds puissants sont équipés de 48 cœurs de calcul, 256 Go de RAM et ont 2 processeurs quadri-processeurs. Les nœuds graphiques ont 16 cœurs de calcul, 64 Go de RAM et sont équipés de cartes NVidia Quadro K4000. En tout, le serveur compte 672 cœurs de calcul, ce qui lui confère une puissance théorique de 8 téraflops. Si l'on inclut les 64 cœurs des nœuds graphiques, la puissance totale atteint 13 téraflops, avec une mémoire disponible de 2,3 téraoctets. Le système de stockage du serveur offre un espace disque de 14 téraoctets.

Librairie	version
Spacy	3.5.2
torch	1.10.1
transformers	4.18.0
tranformer-adapters	4.17.0

Tab. 1.1 : Librairie et version utilisée dans le projet du stage

Pour le stage, nous avons travaillé en utilisant l'environnement de développement VS Code avec une connexion SSH au cluster régional du BRGM. Toutes nos expérimentations ont été lancées sur ce cluster. Pour l'implémentation des modèles, nous avons utilisé le langage Python en version 3.6. Principalement, nous avons utilisé le framework PyTorch pour mettre en œuvre différentes architectures. Les différentes bibliothèques et technologies principales que nous avons utilisées sont répertoriées dans le tableau 1.1. Pour chaque expérimentation, nous avons lancé la tâche sur le serveur avec les configurations spécifiées dans le tableau 1.2.

Noeud	GPU02
RAM	64Go
Nb cpus	8
type de gpu	GPU-intel
Carte graphique	carte NVidia Quadro K4000

Tab. 1.2 : Configuration de lancement des expérimentation sur le Leto

Contrainte technique Lors de notre stage, nous avons rencontré des contraintes techniques liées à l'allocation des ressources sur le cluster de calcul. Étant donné qu'il s'agit d'un cluster régional utilisé par de nombreux utilisateurs avec une forte demande, nous n'avons pas eu l'occasion de tester d'autres configurations pour les modèles présentés dans ce rapport.

5 Synthèse

En conclusion, ce chapitre introductif a permis de présenter l'organisme BRGM qui nous a accueilli pour le stage, en mettant en avant sa mission principale axée sur la gestion des ressources et des risques liés aux sols et sous-sols. Nous avons également souligné l'importance de l'unité des risques et prévention au sein de laquelle notre travail s'est déroulé ainsi que l'organisme collaborateur L'Ecoles de Mines Télécom d'Albi avec son centre de génie industrielle. De plus, nous avons abordé le projet de recherche national RéSoCIO qui constitue le cadre global de notre stage, visant à utiliser les réseaux sociaux de manière opérationnelle pour la gestion des catastrophes naturelles.

Nous avons détaillé la structure et l'organisation de ce projet, en mettant en évidence les principaux lots qui nous intéressent pour notre travail. Ces informations ont permis de situer notre contribution au sein de ce projet d'envergure.

En résumé, ce chapitre introductif a permis de mettre en lumière le secteur d'activité et l'organisation du cadre de notre stage. Il nous a aidé à fixer les objectifs du stage et à avoir une vision claire des attentes qui y sont associées. Dans les chapitres suivants, nous aborderons le cadre scientifique de notre stage, en commençant par une revue de la littérature qui couvrira les aspects techniques et théoriques répondant à nos attentes en matière de travail.

Chapitre II

Revue de la littérature

Dans ce chapitre, nous introduirons le domaine du traitement automatique du langage naturel (TALN) en mettant en évidence les tendances actuelles qui y sont observées. Ensuite, nous procéderons à une revue de la littérature sur les transformers, une architecture qui a révolutionnée le domaine du TALN. Nous présenterons les modèles existants dans la littérature qui pourraient répondre à notre question, en détaillant leurs approches respectives.

1 Le traitement automatique du langage naturel (TALN)

Une langue peut être définie comme un système de règles et de symboles utilisés pour transmettre des informations. Le traitement automatique du langage naturel (TALN) est une branche de l'intelligence artificielle qui vise à permettre aux ordinateurs de comprendre, d'analyser et de générer du langage naturel. Dans cette partie nous allons découvrir l'histoire du TALN et explorer les applications du TALN, ses avancées récentes et ses défis sur la base d'une revue de la littérature. En particulier les éléments abordées dans la publication : "Natural language processing : State of the art, current trends and challenges" [7].

1.1 Histoire du TALN

Dans les années 1940, les recherches sur la traduction automatique (TA) avaient déjà commencé, mais le terme de traitement automatique du langage naturelle (TALN) n'était pas encore utilisé. À cette époque, les recherches sur la TA et le TALN ne se limitaient pas à une région spécifique, mais le russe et l'anglais étaient les langues dominantes pour la recherche sur la TA. Et bien que , selon le rapport ALPAC de 1966, la TA soit considérée comme une impasse, certains systèmes de production de TA fournissaient des résultats aux clients dès le milieu des années 1980.

Au début des années 1960, des travaux sur l'utilisation des ordinateurs pour les études littéraires et linguistiques avaient également commencé. Les travaux emblématiques de l'époque sont représentés par les systèmes BASEBALL Q-A en 1961, suivis de LUNAR et de Winograd SHRDL.

les systèmes BASEBALL Q-A [14], développés en 1961, ont marqué une avancée significative dans les systèmes de question-réponse automatisés. Ces systèmes étaient capables de répondre à des questions en utilisant des bases de connaissances préalablement structurées. Ils ont jeté les bases de la recherche en TALN en permettant une interaction plus sophistiquée avec les ordinateurs.

Par la suite, les travaux ont évolué vers des systèmes plus avancés tels que LUNAR et Winograd SHRDL. LUNAR, développé dans les années 1970, était axé sur la compréhension du langage naturel pour effectuer des tâches spécifiques, comme la traduction automatique. Il a contribué à l'amélioration des capacités linguistiques des systèmes TALN en explorant des techniques de traitement sémantique.

Winograd SHRDL [17], développé également dans les années 1970, s'est concentré sur la résolution de problèmes linguistiques complexes, en mettant l'accent sur les références ambiguës dans le langage naturel. Il a introduit des techniques de représentation de la connaissance et de raisonnement pour mieux traiter ces ambiguïtés.

Ces systèmes, à savoir BASEBALL Q-A, LUNAR et Winograd SHRDL, ont tous contribué à l'avancement des capacités linguistiques et du traitement des tâches dans le domaine du TALN. Leur sophistication croissante a ouvert la voie à de nouvelles perspectives de recherche et de développement dans le domaine, et a permis de repousser les limites de la compréhension et de la génération automatiques de texte.

Dans les années 1980, la théorie grammaticale computationnelle est devenue un domaine de recherche très actif lié à la logique du sens et à la capacité de la connaissance à traiter les croyances et les intentions de l'utilisateur, ainsi que des fonctions telles que l'accentuation et les thèmes. À la fin de la décennie, des processeurs de phrases puissants à usage général tels que le Core Language Engine de SRI et la théorie de la représentation du discours offraient un moyen d'aborder des discours plus étendus dans le cadre grammatico-logique. Cette période a été marquée par la croissance des communautés et la disponibilité de ressources pratiques, de grammaires, d'outils et d'analyseurs syntaxiques.

Dans les années 1990, le traitement statistique du langage est devenu un axe de recherche majeur, avec des recherches sur la désambiguïsation des sens des mots, les réseaux probabilistes et le lexique. L'extraction d'informations et la génération automatique de résumés sont également devenues des domaines de recherche importants. De plus, des recherches sur la modélisation de l'utilisateur et l'interprétation fine ont également été menées.

Au début des années 2000, les recherches en TALN ont continué à progresser, avec des avancées dans des domaines tels que l'apprentissage automatique, l'apprentissage profond et la compréhension du langage naturel. Ces percées ont entraîné une explosion d'applications dans des domaines tels que les chatbots, les assistants virtuels et l'analyse des sentiments.

Le TALN est un domaine qui se consacre à l'interprétation, l'analyse et la manipulation du langage humain à l'aide d'algorithmes, d'outils et de méthodes. Cependant, la complexité et la variabilité du langage naturel posent des défis importants pour atteindre ces objectifs avec une seule approche. Par conséquent, les chercheurs se sont concentrés sur le développement d'une gamme diversifiée d'outils et de méthodes dans le domaine du TALN.

Parmi les premiers développements en TALN concernant la modélisation neuronale du langage et les techniques d'embedding de mots, on trouve les réseaux neuronaux, tels que les réseaux neuronaux à convolution (CNN) [10] et les réseaux neuronaux récurrents (RNN) [19], ont joué un rôle essentiel dans la transformation du TALN. Ces réseaux ont permis des tâches telles que la classification de phrases, l'analyse de sentiment, la résumé de texte, la traduction automatique, et bien d'autres. Les mécanismes d'attention et les modèles de transformers ont également fait progresser le domaine en permettant aux réseaux d'apprendre à quoi prêter attention et de capturer des dépendances à plus long terme.

Parallèlement au développement des réseaux neuronaux, divers outils ont été créés pour aborder des tâches spécifiques en TALN. Les outils d'analyse de sentiment visent à extraire les sentiments sur des sujets donnés en analysant les documents à la recherche de mots positifs et négatifs. Les étiqueteurs de parties du discours (POS) classifient les mots en catégories telles que les noms, les adjectifs et les verbes, et ont été développés pour différentes langues. Le "chunking", ou "Shadow Parsing", permet de séparer des phrases significatives à partir de texte non structuré. Les outils de reconnaissance d'entités nommées (NER) identifient et catégorisent des entités spécifiques telles que les lieux, les personnes et les organisations. Les outils de détection des émotions analysent la parole, les expressions faciales, les gestes et le texte pour identifier différents types d'émotions. L'étiquetage sémantique des rôles (SRL) attribue des rôles sémantiques aux

phrases, permettant une compréhension plus approfondie des relations entre les mots et leurs fonctions.

Ces outils et avancées en TALN ont ouvert la voie à des applications plus larges dans divers domaines. Le TALN est utilisé dans l'analyse de sentiment, l'extraction d'informations, les systèmes de questions-réponses, les chatbots, les assistants virtuels, et bien plus encore. La capacité à analyser et traiter le langage humain a révolutionné les processus de communication, d'analyse de données et de prise de décision. En abordant les défis de la variabilité du langage naturel et en exploitant la puissance des réseaux neuronaux et des outils spécialisés, le TALN continue d'évoluer et de façonner notre manière d'interagir avec le langage et l'information.

1.2 Applications du TALN

Le TALN est une technologie polyvalente qui trouve des applications dans divers domaines, y compris la traduction automatique, la détection du spam par courrier électronique, l'extraction d'informations, le résumé et la réponse aux questions. Dans ce texte, nous explorons certains applications.

- **Traduction automatique** : consiste à traduire des phrases d'une langue à une autre en utilisant des moteurs statistiques. Le défi est de préserver le sens et la grammaire des phrases. Les systèmes de traduction automatique recueillent des données parallèles entre les langues et utilisent l'apprentissage automatique statistique pour déterminer la probabilité de correspondance entre langues. Google, par exemple, a introduit un système de traduction automatique basé sur des réseaux neuronaux artificiels et l'apprentissage profond.

- **Classification du texte** : Les systèmes de classification de texte permettent de trier de grandes quantités de données en différentes catégories prédéfinies. Cela peut inclure la classification de documents officiels, de données de marché, d'articles d'actualité, voire le filtrage des courriers indésirables (spam) par e-mail. Différents types de filtres anti-spam, tels que les filtres de contenu, les filtres d'en-tête et les filtres d'autorisation, sont utilisés pour identifier et bloquer les e-mails non désirés. Des techniques d'apprentissage automatique, telles que l'apprentissage de règles, la classification naïve bayésienne et les machines à vecteurs de support, sont appliquées à la classification de texte et au filtrage du spam.

- **Extraction d'informations** : L'extraction d'informations implique l'identification de phrases ou d'entités importantes dans les données textuelles. Il joue un rôle vital dans des tâches telles que le résumé, la construction de bases de données et la classification des éléments de texte. Les modèles Markov cachés (HMM) sont utilisés pour extraire des champs pertinents des documents de recherche, permettant une recherche et une correspondance efficaces des références. Des techniques telles que les étiquetage morpho-syntaxique, le chunking, la suppression des mots d'arrêt, le stemming et la désambiguation du sens du mot aident à l'extraction d'informations.

- **Résumé du texte** : La synthèse de texte est très importante pour gérer le surplus d'informations. Elle permet de réduire de grandes quantités de données tout en pré-

servant leur sens. Le résumé peut être réalisé sur un seul document ou sur plusieurs documents, et il peut être général ou centré sur une requête spécifique. Pour ce faire, différentes techniques sont utilisées, qu'elles soient supervisées ou non supervisées. Parmi celles-ci, on retrouve les modèles thématiques basés sur des phrases bayésiennes, la factorisation avec des bases de données et la génération de résumés centrés sur les aspects thématiques.

Ce ne sont que quelques exemples montrant les applications du TALN dans différents domaines, mettant en évidence l'importance du TALN pour le traitement et l'extraction d'informations significatives à partir de données textuelles.

1.3 Enjeux du TALN

Les applications du TALN connaissent une croissance constante, mais de nouveaux défis émergent malgré les avancées réalisées récemment. Certains défis courants comprennent :

- **Ambiguïté contextuelle** : Les mots et expressions dans le langage peuvent avoir différentes significations en fonction du contexte, ce qui pose un défi pour les systèmes de TALN. Alors que les humains peuvent facilement comprendre ces nuances, concevoir des algorithmes pour interpréter et traiter de telles variations devient difficile. De même, traiter les synonymes dans le langage représente un autre défi, car les humains utilisent différents mots pour exprimer la même idée.
- **Homonymes** : Les homonymes sont des mots qui se prononcent de la même manière mais qui ont des significations différentes. Ils posent des défis pour les applications de question-réponse et de transcription vocale et l'extraction de l'information notamment lorsque les mots ne sont pas sous forme écrite.
- **Sarcasme et ironie** : Les phrases contenant du sarcasme et de l'ironie peuvent être interprétées à l'opposé par les humains. Concevoir des modèles capables de comprendre et de gérer de telles phrases de manière précise est une tâche complexe en TALN.
- **Ambiguïté** : Le langage peut comporter des phrases pouvant être interprétées de plusieurs manières, ce qui entraîne de l'ambiguïté. Obtenir une précision plus élevée dans l'interprétation de telles phrases est un domaine qui nécessite des améliorations supplémentaires.
- **Langage informel et expressions spécifiques à la culture** : Les phrases informelles, les expressions, les idiomes et le langage spécifique à une culture rendent difficile la conception de modèles de TALN destinés à une utilisation générale. Bien que l'entraînement sur un grand ensemble de données puisse améliorer les performances des modèles, il reste difficile de tenir compte des mots ayant des significations différentes dans différentes régions géographiques.
- **Défis spécifiques à un domaine** : Traiter différents domaines tels que l'éducation, la santé, le droit et la défense introduit des variations dans la signification des mots

et des phrases. Les modèles conçus pour des domaines ou des régions géographiques spécifiques peuvent ne pas bien fonctionner dans des applications plus larges, ce qui nécessite des solutions pour relever ces défis.

- **Mots mal orthographiés et mal utilisés** : Les mots incorrectement orthographiés ou mal utilisés peuvent poser problème, notamment lorsqu'il s'agit de prédire l'intention de l'écrivain dans un domaine ou une région géographique spécifique. Comprendre le sarcasme, les expressions et les phrases informelles ajoute une complexité supplémentaire.

Bien que les modèles de TALN aient réalisé des progrès significatifs pour les langues les plus couramment utilisées, il est nécessaire de développer des modèles adaptés aux préférences linguistiques et technologiques des utilisateurs individuels, plutôt que de s'appuyer sur des connaissances spécifiques à une langue particulière.

1.4 Conclusion

Dans cette partie on a cité les différents points d'évolutions du TALN dans le temps, et les principaux cas d'utilisations, dans la partie suivante on introduira une vue de l'Etat de l'art sur les *Transformers*, qui ont mis en révolution le TALN avec ses différents enjeux.

2 les *Transformers*

Le monde du TALN a été révolutionné par les Transformers. Cette architecture novatrice ne cesse d'émerveiller les chercheurs depuis sa création en 2017. Elle est utilisée dans une variété de tâches de traitement du langage naturel et offre des résultats exceptionnels pour la traduction automatique, la classification de texte, la génération de langue naturelle et bien plus encore. Dans cette partie, nous explorons comment fonctionne cette architecture transformer pour différentes tâches de TALN.

2.1 Le mécanisme d'Attention

Le mécanisme d'attention est un outil crucial pour permettre aux modèles de machine learning de comprendre le texte contextuel. Cela leur donne la capacité de se concentrer sur des parties spécifiques du texte et de les traiter en profondeur. Comme l'a souligné Hewitt [4], "Les modèles de machine learning peuvent se concentrer sur des parties spécifiques du texte grâce à l'utilisation du mécanisme d'attention". Ce processus garantit que les algorithmes apprennent à identifier les informations importantes dans le texte, plutôt que de simplement répéter des mots-clés sans signification. Lorsque ce mécanisme est utilisé efficacement, il peut aider à résoudre certains problèmes courants associés au traitement automatique du langage naturel. Par exemple, il peut être difficile pour une machine d'identifier les pronoms et leurs références dans un texte.

En somme, le mécanisme d'attention joue un rôle essentiel dans la compréhension et le traitement précis des textes contextuels par les modèles de machine learning. Il offre une

méthode plus sophistiquée qui permet aux algorithmes non seulement d'apprendre mais aussi de bien interpréter ces textes complexes tout comme Hewitt [4] a mentionné : "Pour cela ils peuvent facilement se focaliser sur certaine partie nécessaires."

L'auto-attention, une technique récente dans le domaine de l'apprentissage automatique [21], permet aux modèles de comprendre les relations entre les mots dans une phrase. Cette technique consiste à donner plus d'importance aux mots qui ont un lien sémantique fort avec les autres et compare tous les membres de la séquence d'entrée entre eux et modifie les positions correspondantes de la séquence de sortie. En d'autres termes, la couche d'auto-attention effectue une recherche différentielle par valeur-clé dans la séquence d'entrée pour chaque entrée et ajoute les résultats à la séquence de sortie. L'auto-attention permet au modèle d'évaluer l'importance des différents éléments d'une séquence d'entrée et d'ajuster dynamiquement leur influence sur la sortie. Le mécanisme d'auto-attention est utilisé pour capturer les informations contextuelles de la séquence d'entrée, ce qui est particulièrement important pour les tâches de traitement du langage, où la signification d'un mot peut changer en fonction de son contexte dans une phrase ou un document. Le mécanisme d'auto-attention fait partie de l'architecture neuronale qui permet au modèle de mettre en évidence de manière dynamique les caractéristiques pertinentes des données d'entrée.[4]

2.2 L'architecture des *Transformers*

Le Transformer est une architecture (Figure 2.1) de réseau de neurones qui a été introduite pour la première fois en 2017 par des chercheurs du Google dans la publication "Attention is all you need !" [21]. Son architecture se compose de deux éléments principaux : l'encodeur et le décodeur.

L'encodeur fait correspondre une séquence d'entrée à une séquence de représentations continues, tandis que le décodeur reçoit la sortie du codeur ainsi que la sortie du décodeur au pas de temps précédent pour générer une séquence de sortie. L'architecture du *Transformer* est basée sur un mécanisme d'auto-attention (Self-attention) qui pondère différemment l'importance de chaque partie des données d'entrée. Le mécanisme d'auto-attention se compose de plusieurs éléments, notamment l'attention au produit de points mis à l'échelle, l'attention à plusieurs têtes et l'encodage positionnel.

- **Scaled dot product attention** L'attention par produit de points échelonnés calcule le produit de points du vecteur de requête et du vecteur clé , Les vecteurs de requête représentent des informations que nous souhaitons extraire ou obtenir plus d'attention, tandis que les vecteurs clés contiennent des informations importantes avec lesquelles nous voulons comparer les vecteurs de requête. En multipliant chaque vecteur de requête avec tous les vecteurs clés, nous mesurons la similarité entre les éléments de la séquence d'entrée. Les scores d'attention mis à l'échelle sont ensuite utilisés pour pondérer les vecteurs de valeurs, permettant ainsi d'obtenir des représentations contextuelles et d'effectuer des opérations d'attention efficaces et parallèles.

- **Multi-Head Attention** est une variante de l'attention par produit de points mis à

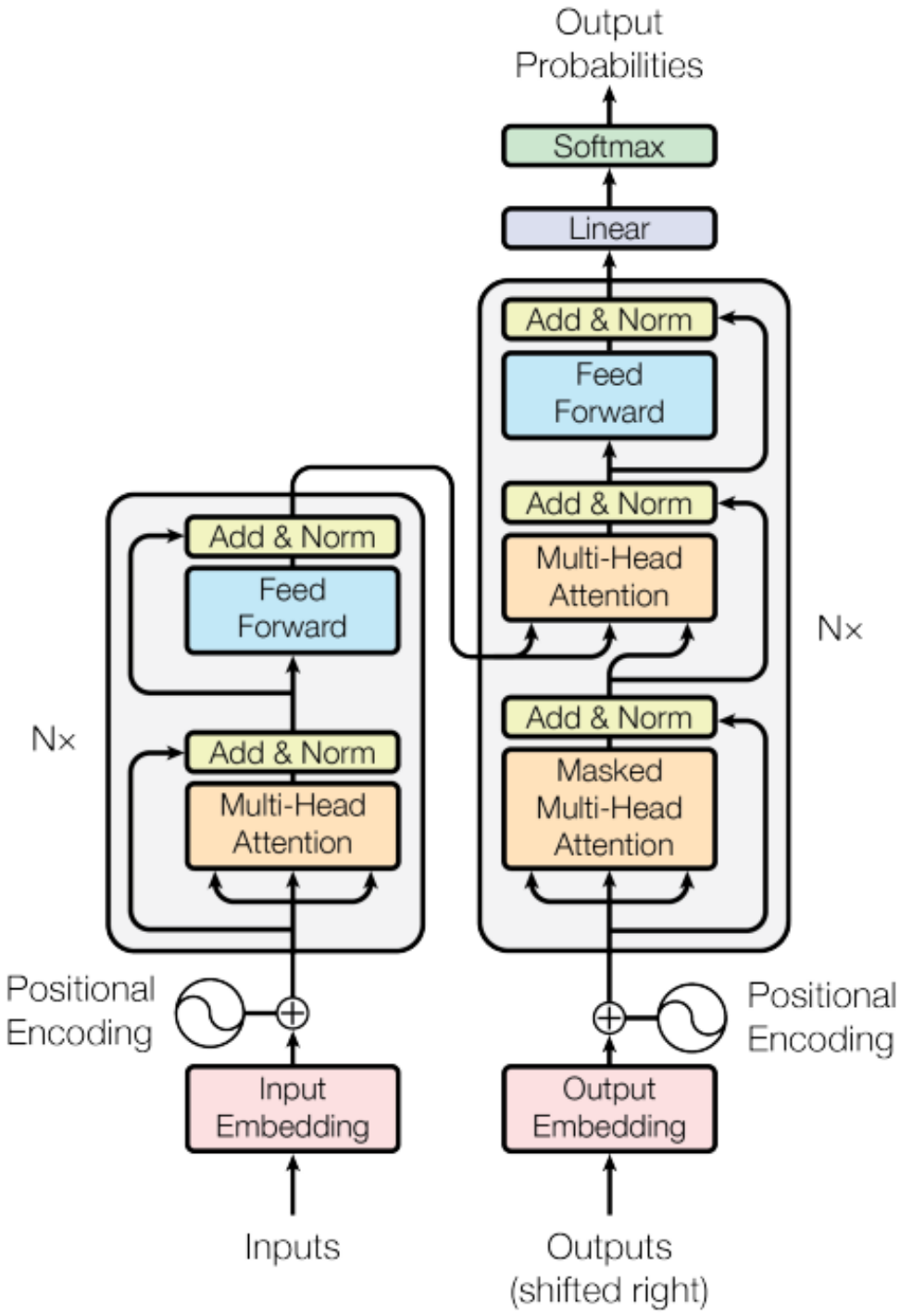

Fig. 2.1 : L'architecture du Transformer [21]

l'échelle qui permet au modèle de prendre en compte conjointement des informations provenant de différents sous-espaces de représentation à différentes positions

- **Positionnal encoding** est utilisé pour injecter des informations positionnelles dans les encastrements d'entrée, ce qui aide le modèle à préserver les informations positionnelles de la séquence d'entrée.

L'architecture Transformer présente plusieurs avantages par rapport aux algorithmes d'apprentissage profond traditionnels tels que les réseaux neuronaux récurrents (RNN). Elle a notamment la capacité de gérer les dépendances à longue portée et à traiter et former plus de données en moins de temps.Ces spécificités lui ont permis d'être utilisée dans une variété de tâches différentes et de devenir l'une des architectures les plus populaires pour le traitement du langage naturel (TALN). Une augmentation significative des performances sur plusieurs tâches TALN , a, en effet, été observée depuis l'introduction du Transformer. Cela suggère que cette architecture offre un moyen efficace d'améliorer les résultats sur ces tâches sans nécessiter beaucoup d'expertise humaine ou technique supplémentaire. De plus, la flexibilité inhérente au Transformer signifie qu'il peut être utilisé dans plusieurs domaines où il n'a pas encore été exploré pleinement. Il pourrait également offrir des avantages considérables pour certaines applications industrielles telles que les chatbots. Enfin, certains chercheurs se sont également intéressés à comprendre pourquoi cette méthode fonctionne si bien par rapport à certains autres modèles existants basés sur la récurrence comme les LSTM (Long Short-Term Memory), une question qui continue a être traitée dans les travaux de recherche à venir.

2.3 Les adaptateurs pour les *Transformers*

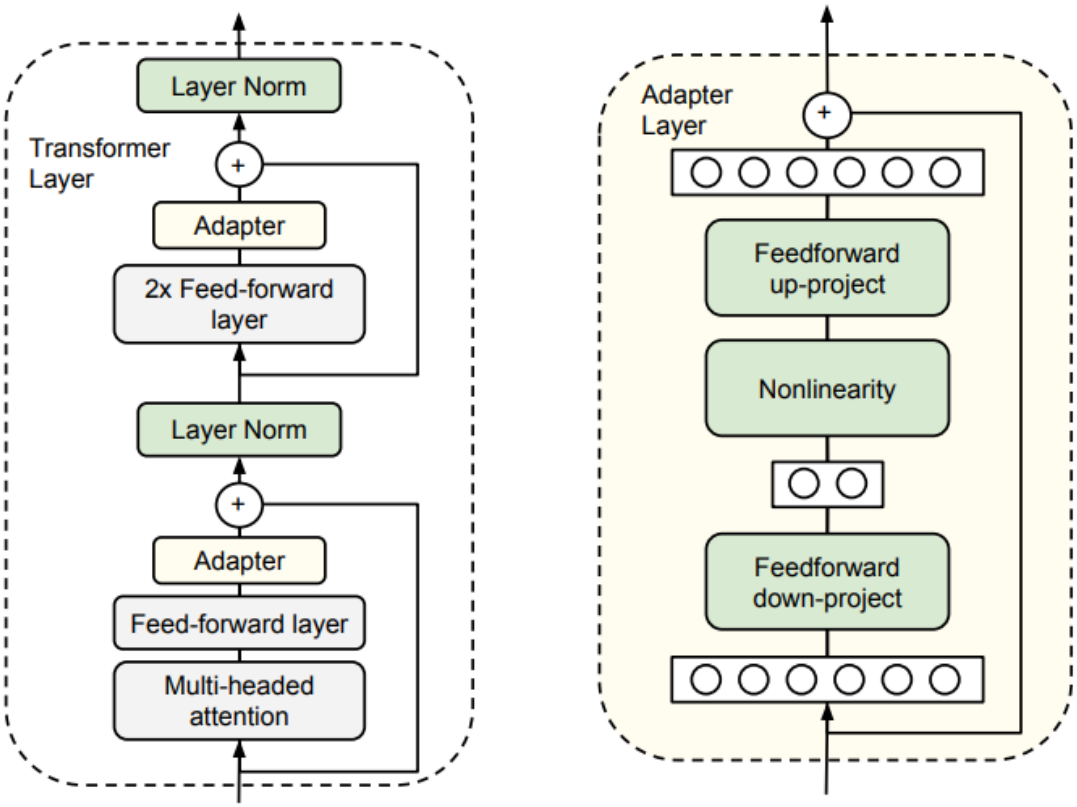

Fig. 2.2 : Exemple d'un adaptateur *Transformer*

Le Transfer Learning est un processus clé pour améliorer les performances des modèles de TALN. Cependant, ce processus peut être coûteux en termes de temps et de ressources

de calcul. Heureusement, l'utilisation d'adaptateurs pour les *Transformers* a été introduite comme une méthode plus efficace pour le transfert d'apprentissage dans les modèles TALN [5]. Les adaptateurs sont des couches supplémentaires (exemple dans la figure 2.2) qui peuvent être entraînées sur des tâches spécifiques et connectées à un modèle existant sans modifier ses poids internes. Cette approche permet aux utilisateurs de réutiliser facilement leurs modèles existants sur différentes tâches avec moins d'effort en évitant la nécessité de finetuner entièrement leur modèle ou même le re-entraîner à partir de zéro. En outre, l'utilisation d'une technique appelée "partage pondéral" [16] peut augmenter encore davantage l'efficacité et la performance des adaptateurs en partageant leurs poids entre plusieurs tâches distinctes grâce à l'utilisation du même vocabulaire axé sur la langue naturelle dans différents contextes linguistiques. L'étude menée par ces auteurs [5] a permis ainsi démontré que cette méthode était capable de rivaliser avec celle consistant à fine-tuner complètement tout le modèle standard lorsqu'elle était correctement entraînée. Dans l'ensemble, utiliser des adaptateurs représente donc une solution prometteuse pour améliorer significativement le transfert d'apprentissage entre différentes tâches liées au traitement du langage naturel. Les résultats récents montrent que cette approche est capable d'obtenir des performances similaires, voire meilleures, par rapport à des méthodes plus traditionnelles comme le fine-tuning complet ou l'entraînement.

2.4 Les *Transformers* pré-entrainé

Les *Transformers* pré-entrainés sont un type de grand modèle de langage (LLM). Ils sont pré-entraînés sur de grandes quantités de données et peuvent être affinés pour des tâches spécifiques. Les *Transformers* pré-entraînés peuvent réduire les coûts de calcul et la quantité de données nécessaire à l'apprentissage. Ils sont disponibles par le biais d'API et d'outils fournis comme la bibliotheque python Transformers le hub de HuggingFace [1].

BERT

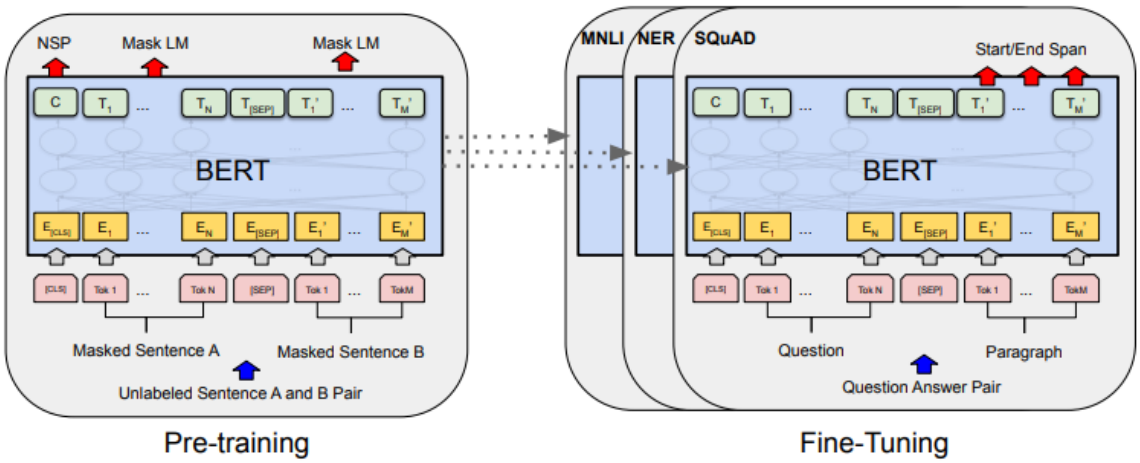

FIG. 2.3 : Procédures globales d'entrainement et de fine tuning pour BERT

[1] lien du transformer-hub du HuggingFace : https://huggingface.co/docs/transformers/

Bert, abréviation de Bidirectional Encoder Representations from Transformers [2], est un modèle d'embedding de texte introduit en 2018 par Jacob Devlin et ses collègues de Google AI Language. BERT utilise des transformers, une architecture de réseau de neurones basée sur l'attention, pour capturer la sémantique contextuelle des mots dans un texte. BERT est actuellement l'un des modèles les plus performants pour de nombreuses tâches de TALN, notamment la classification de texte, la reconnaissance d'entités nommées, la traduction automatique et la génération de texte. Selon Devlin et al, Bert a été conçu pour comprendre le contexte des mots dans une phrase en analysant le texte environnant et en tenant compte des nuances telles que les expressions idiomatiques ou l'ambiguïté des mots. Cette approche s'écarte considérablement des algorithmes traditionnels d'apprentissage automatique qui s'appuient sur des règles et des caractéristiques prédéfinies. En utilisant des architectures basées sur des *Transformers*, Bert peut traiter des phrases entières au lieu de s'appuyer uniquement sur des mots individuels. Il peut ainsi s'attaquer à des tâches complexes telles que la réponse à des questions ou l'analyse de sentiments avec une précision remarquable. Cependant, si Bert représente une étape importante dans le développement de l'IA, son succès met également en évidence certaines limites des approches actuelles du traitement du langage naturel. Par exemple, les critiques affirment qu'en se concentrant trop sur les modèles de transformation à grande échelle comme Bert, on risque de négliger d'autres domaines de recherche tels que le raisonnement multimodal ou la compréhension du sens commun. En outre, ces modèles nécessitent de grandes quantités de données étiquetées et de puissance de calcul pour être entraînés efficacement - des ressources qui ne sont pas toujours disponibles en dehors des cercles académiques.

RoBERTa

RoBERTa [12] est un modèle de Transformer pré-entrainé comme BERT. Cependant, RoBERTa présente quelques différences clés par rapport à BERT. Tout d'abord, RoBERTa a été entraîné sur un ensemble de données plus vaste et varié que BERT, ce qui lui permet de mieux saisir les subtilités du langage. De plus, RoBERTa a une très bonne capacité de compréhension des relations contextuelles entre les mots puisqu'il utilise une méthode appelée "masking dynamique", où les mots sont masqués de manière aléatoire à chaque itération d'entraînement. Sonpré-entraînement est plus long,et durant cette phase le modèle est exposé à de vastes quantités de texte non étiqueté pour apprendre les structures linguistiques sous-jacentes. Par la suite, le modèle est fine-tuné sur des tâches spécifiques en utilisant des données étiquetées, ce qui permet d'adapter ses connaissances à des problèmes particuliers. Cette approche permet à RoBERTa d'obtenir des résultats plus précis et plus fiables que Bert sur des tâches de TALN.

Bien que RoBERTa soit un modèle puissant, il présente également certaines limitations. Tout d'abord, en raison de sa taille et de sa complexité, RoBERTa nécessite une puissance de calcul importante et des ressources considérables pour être entraîné. Cela signifie qu'il peut être difficile pour les chercheurs et les développeurs de l'utiliser et de l'adapter à leurs besoins spécifiques. De plus, RoBERTa peut rencontrer des difficultés avec les tâches qui nécessitent une compréhension de connaissances spécifiques ou qui nécessitent une logique complexe, car il se base principalement sur les relations statistiques entre les mots plutôt

que sur une compréhension profonde du sens.

En résumé, RoBERTa est un modèle de langage amélioré par rapport à BERT, grâce à son entraînement sur un ensemble de données plus vaste et varié, son utilisation de la méthode du masking dynamique et son optimisation de l'apprentissage. Cependant, il présente également des limitations liées à ses exigences en termes de ressources, sa capacité limitée à comprendre les connaissances spécifiques et son potentiel de biais lié aux données d'entraînement. Il est important de prendre en compte ces limitations lors de l'utilisation de RoBERTa dans des applications réelles.

CamemBERT

CamemBERT [13] est un modèle de texte embedding pour la langue française qui est basé sur l'architecture RoBERTa, qui est une variante améliorée de l'architecture BERT. Développé par des chercheurs de l'Université Paris-Saclay, CamemBERT est actuellement l'un des meilleurs modèles de TALN pour le français, capable de surpasser les modèles existants pour de nombreuses tâches de TALN dans cette langue. Il est entraîné sur un corpus de textes français de grande taille, qui comprend des textes de diverses sources, notamment des journaux, des livres, des sites web et des forums. Ce corpus est pré-traité et utilisé pour entraîner le modèle de texte embedding, qui peut ensuite être utilisé pour une variété de tâches de TALN en français.

Les performances de CamemBERT ont été évaluées pour plusieurs tâches de TALN en français, notamment la classification de texte, la reconnaissance d'entités nommées, la traduction automatique et la génération de texte.

En utilisant CamemBERT pour le français, il est donc possible de bénéficier des avantages des modèles de text embedding pour améliorer la précision et la pertinence des résultats de recherche, des traductions, des résumés, des réponses automatiques, etc.

2.5 Conclusion

Dans cette partie on a exploré l'architecture qui a révolutionné le TALN et montré sesdifférentes composantes. L'utilisation du *Transformers* dans le cadre du travail réalisé dans ce stage pourra nous aider à générer des représentations vectorielles du texte en respectent leur dimension contextuelle par le mécanisme d'attention et en adaptant un *Transformer* pré-entrainé pour nos taches en utilisant le Transfer Learning.

3 Etat de l'art

3.1 La ségmentation du texte

Le clustering est une technique d'analyse de données non supervisée qui vise à regrouper des données similaires en ensembles homogènes appelés clusters. Cette technique est largement utilisée dans de nombreux domaines, y compris le traitement de texte, où elle est souvent utilisée pour regrouper des documents similaires en fonction de leur contenu.

Le clustering des textes consiste à organiser et structurer de grandes quantités de données textuelles en groupes ou en clusters, afin d'identifier des groupes de documents qui partagent des similitudes dans leur contenu. Cette technique peut être utilisée pour diverses tâches, telles que l'analyse de tendances, l'analyse de sentiment, la classification de documents, etc.

Les techniques de clustering de texte classique sont basées sur des méthodes statistiques et probabilistes qui utilisent des représentations vectorielles des mots et des documents pour regrouper les documents similaires en clusters. Cependant, avec la croissance explosive des données textuelles, de nouvelles techniques de clustering basées sur les graphes ont émergé dans l'état de l'art pour surmonter les limites des méthodes de clustering classiques. Ces techniques de clustering basées sur les graphes exploitent les relations sémantiques entre les mots et les documents pour découvrir des structures cachées dans les données textuelles et améliorer la qualité des clusters obtenus. Dans cette optique, il est important de comprendre l'état de l'art des techniques de clustering de texte classique et basé sur les graphes pour améliorer l'efficacité et la qualité de l'analyse de données textuelles.

3.2 Approches basées sur les graphes

Les graphes sont largement utilisés dans le traitement automatique du langage naturel [6] (TALN) pour modéliser les relations sémantiques entre les entités dans un texte. Les graphes peuvent être utilisés pour représenter les relations entre des mots, des phrases, des entités nommées, des concepts, des événements, etc. Dans cette partie, nous allons nous concentrer sur deux applications des graphes dans le TALN : la prédiction de liens et la détection de communautés.

La prédiction de liens [11] (ou link prediction) est une tâche qui consiste à prédire la présence ou l'absence d'un lien entre deux nœuds dans un graphe. Cette tâche est souvent utilisée pour compléter un graphe incomplet en prédisant les liens manquants. Toutefois, cette méthode n'est pas toujours adaptée au TALN, car elle suppose que tous les liens ont la même importance, ce qui n'est pas le cas pour les relations sémantiques entre les entités d'un texte. En revanche, la détection de communautés (ou community detection) peut être une technique efficace pour modéliser les relations sémantiques entre les entités. Cette méthode consiste à identifier des groupes de nœuds (ou communautés) qui sont plus densément connectés entre eux qu'avec le reste du graphe. Ces communautés peuvent correspondre à des groupes de mots qui ont une signification sémantique similaire ou à des groupes d'entités nommées qui sont liées par des relations sémantiques spécifiques.

La détection de communautés [3] peut être appliquée à différentes tâches du TALN, telles que l'analyse de sentiments, l'extraction d'informations, la classification de textes, etc. Par exemple, dans l'analyse de sentiments, la détection de communautés peut être utilisée pour identifier des groupes de mots qui expriment des émotions similaires (par exemple, des mots qui expriment la colère, la tristesse, la joie, etc.). Dans l'extraction d'informations, la détection de communautés peut être utilisée pour identifier des groupes d'entités nommées qui sont liées par des relations sémantiques spécifiques (par exemple, des personnes, des lieux, des organisations, etc.).

En résumé, les graphes sont des outils puissants pour modéliser les relations sémantiques

entre les entités dans un texte. La prédiction de liens peut être utile dans certaines applications, mais elle n'est pas toujours adaptée au TALN. En revanche, la détection de communautés peut être une technique efficace pour identifier des groupes d'entités qui ont une signification sémantique similaire et qui sont liées par des relations sémantiques spécifiques.

La méthode CLARE

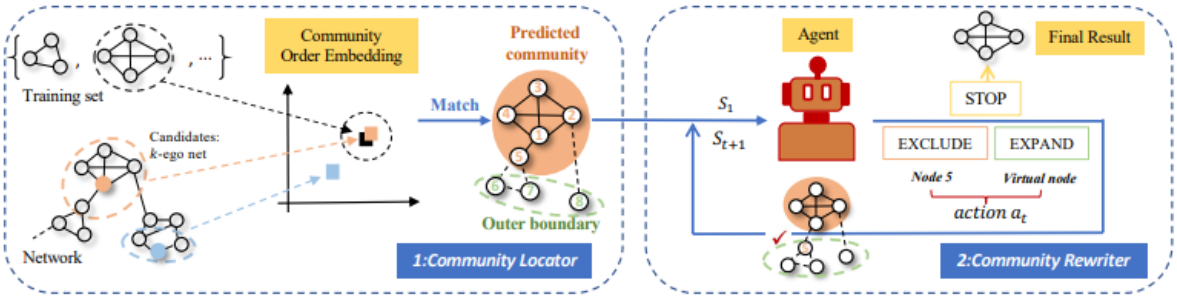

FIG. 2.4 : l'architecture CLARE

CLARE (Community Detection Algorithm for Semi-Supervised Learning) [22] est une méthode de détection de communautés semi-supervisée qui utilise des informations à la fois non marquées comme les informations relatives à la fréquence des mots et les métadonnées et partiellement marquées comme le contexte et la sémantique du texte pour détecter des communautés dans un réseau. La méthode a été proposée par Wu et al [22] en 2022.

La méthode CLARE commence par attribuer à chaque nœud dans le réseau une étiquette de communauté initiale. Ensuite, elle utilise une fonction de coût qui prend en compte à la fois l'information de la structure du réseau et les informations partiellement marquées pour estimer la qualité des attributions de communauté actuelles. Cette fonction de coût est utilisée pour ajuster les étiquettes de communauté pour chaque nœud, de manière à maximiser la qualité globale des attributions de communauté dans le réseau.

Cette approche peut être appliquée pour la détection d'événements similaires dans les tweets en utilisant un réseau de tweets. Le réseau peut être construit en utilisant des mesures de similarité entre les tweets, par exemple, en utilisant des techniques de traitement du langage naturel (TALN) pour identifier les tweets qui discutent du même sujet ou événement.

Ensuite, elle peut être utilisée pour détecter des communautés dans le réseau de tweets en utilisant des informations à la fois non marquées (les mesures de similarité entre les tweets) et partiellement marquées (par exemple, les hashtags associés aux tweets). Les étiquettes de communauté initiales peuvent être définies à partir des hashtags, et la méthode CLARE peut être utilisée pour ajuster ces étiquettes en utilisant les mesures de similarité entre les tweets. Les communautés résultantes peuvent être interprétées comme des groupes de tweets qui discutent du même événement ou sujet.

Dans le contexte de la détection d'événements similaires dans les tweets, l'utilisation de la méthode CLARE peut également présenter des limites. Les tweets ont un contenu limité et

sont souvent sujets à l'ambiguïté et à la variabilité. Cela peut rendre difficile l'identification précise des événements similaires à partir des tweets. De plus, les tweets sont souvent limités à 280 caractères, ce qui peut limiter la quantité d'informations disponibles pour la détection de communautés. De plus, l'utilisation de hashtags peut également être limitée dans certains contextes, tels que les événements en temps réel, où les hashtags peuvent ne pas être largement utilisés ou n'ont pas encore été créés.

Par conséquent, l'utilisation de hashtags peut ne pas fournir suffisamment d'informations pour la détection de communautés précises. Enfin, la méthode CLARE peut être sensible aux biais dans les tweets, ce qui peut affecter la qualité des résultats et limiter la capacité à interpréter les communautés détectées de manière significative. En somme, bien que la méthode CLARE puisse être utile pour la détection d'événements similaires dans les tweets, elle présente des limites qu'il convient de prendre en compte pour une utilisation efficace et précise.

3.3 Approches basées sur la ségmentation classique

La méthode SCCL

La méthode Support Clustering with Contrastive Learning (SCCL) [23] est une technique innovante pour regrouper des courts textes en utilisant la similarité et la dissimilarité des paires de phrases. Cette méthode permet d'obtenir des résultats efficaces et précis dans le traitement de données textuelles, car elle combine l'apprentissage supervisé avec l'apprentissage non-supervisé. Selon Zhang et al. (2021) [23], cette approche utilise un modèle de prédiction qui apprend à distinguer les paires de phrases similaires des paires de phrases différentes. La méthode Support Clustering via Contrastive Learning prend donc en compte les similitudes entre les phrases tout comme leurs différences pour former un groupe logique. Cette technique représente une avancée significative dans le domaine du traitement automatique du langage naturel, où la segmentation des documents est souvent difficile à réaliser avec précision. En effet, cette approche permettrait d'améliorer la qualité du clustering en réduisant au minimum les erreurs liées aux ambiguïtés sémantiques ou syntaxiques présentes dans le texte. En somme, grâce à sa capacité à combiner différents types d'apprentissages, la méthode Support Clustering via Contrastive Learning se positionne désormais comme une option viable pour les experts cherchant à améliorer leur compréhension de grandes quantités de données textuelles complexes rapidement et efficacement.

Le short text clustering est un défi complexe en raison du manque d'informations contextuelles et sémantiques disponibles. Cependant, la méthode SCCL offre une approche prometteuse pour surmonter ces défis en découvrant des relations sémantiques et syntaxiques plus précises entre les phrases. Cette méthode utilise un réseau de neurones profond qui apprend à associer des paires de phrases similaires ou différentes en fonction de leurs caractéristiques communes. Selon Li et al., cette méthode permet non seulement d'améliorer l'exactitude du clustering, mais également d'obtenir des résultats plus rapidement que les méthodes traditionnelles. En effet, elle peut être utilisée pour résoudre divers problèmes tels que la partition automatique du corpus dans le domaine médical ou littéraire. Par conséquent, cette technique a été largement adoptée dans différents domaines scientifiques

où l'on travaille avec des textes courts nécessitant une analyse rapide sans compromettre la qualité. Ainsi, il est clair qu'en utilisant SCCL comme méthode pour le clustering de textes courts on obtient non seulement des résultats rapides mais aussi précis ce qui ouvre ainsi plusieurs perspectives intéressantes pour notre projet.

4 Synthèse

Dans ce chapitre, nous avons commencé par passer en revue la littérature relative au Traitement Automatique des Langues (TAL). Cette revue a permis de mettre en lumière les différents axes de développement de ce domaine, ses multiples applications et les défis qu'il soulève. Nous avons notamment abordé l'évolution des approches de TAL, depuis les méthodes traditionnelles jusqu'aux méthodes de pointe comme les *Transformers*.

L'architecture des Transformers, qui a considérablement transformé le domaine du TAL, a fait l'objet d'une étude approfondie. En présentant ces modèles qui sont pré-entrainés, nous avons discuté la technique de *Transfer Learning* qui permettera d'adapter ces *Transformers* pré-entrainés à de nouvelles tâches.

Enfin, la dernière partie de ce chapitre a été consacrée à l'examen des méthodes existantes pour le regroupement de textes similaires. Nous avons exploré en détail deux approches principales : une approche basée sur les graphes (CLARE) et une autre basée sur le clustering (SCCL).

La méthode CLARE, qui s'appuie sur une structure de graphe, présente des avantages significatifs. Cependant, elle semble mal adaptée à notre problématique. En effet, l'utilisation de l'apprentissage par renforcement qu'elle requiert est très consommatrice en ressources techniques. De plus, cette méthode nécessite de nombreuses informations spécifiques, comme les métadonnées des utilisateurs de réseaux sociaux, qui ne sont pas disponibles dans nos bases de données. Enfin, le fait que la méthode CLARE ait été testée principalement sur des articles et non sur des tweets limite son utilité dans notre contexte.

Par contre, la méthode SCCL, basée sur le clustering, semble plus adéquate pour notre problématique. Testée sur des tweets, elle utilise des techniques innovantes pour améliorer le regroupement des textes. De plus, elle ne requiert pas de grandes quantités de données puisqu'elle utilise une technique d'augmentation de texte. Par conséquent, nous avons choisi de mettre en œuvre la méthode SCCL pour notre problème et d'expérimenter avec cette architecture.

Chapitre III

Présentation des bases de données

Dans ce chapitre, nous commencerons par une brève introduction sur Twitter, qui est la source de nos données. Ensuite, nous procéderons à une analyse exploratoire des trois jeux de données que nous utiliserons par la suite dans nos expérimentations. Nous mettrons en évidence le processus de préparation de ces données, ainsi que leurs différentes caractéristiques statistiques et sémantiques.

1 Introduction au Twitter

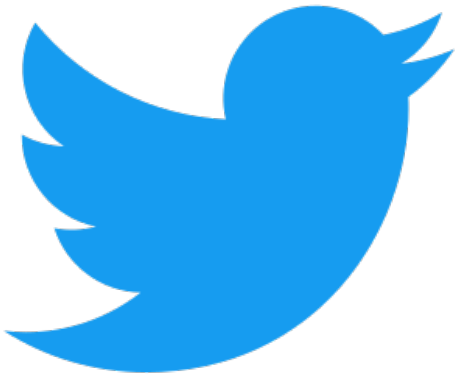

Fig. 3.1 : Logo du Twitter

Twitter est une plateforme de médias sociaux qui permet aux utilisateurs de publier et d'interagir avec des messages courts appelés "tweets". Ces tweets, limités à 280 caractères, peuvent contenir du texte, des images, des vidéos et des liens. Les utilisateurs peuvent suivre d'autres comptes pour voir leurs tweets dans leur flux d'actualités et interagir avec eux en aimant, retweetant et répondant à leur contenu. Connue pour sa nature en temps réel et publique, Twitter est couramment utilisée pour partager des actualités, des opinions et engager des conversations sur divers sujets. De plus, selon "*Campaigning on Twitter : The Handbook for NGOs, Politics & Public Service*" [20], Twitter s'est révélé être un outil puissant pour les campagnes politiques, les organisation non gouvernemental et les organisations de service public.

1.1 L'API du Twitter

L'API de Twitter est un outil puissant qui permet aux développeurs d'accéder aux données de Twitter et d'interagir avec elles. Elle offre plusieurs fonctionnalités pour récupérer des tweets, des profils d'utilisateurs, des tendances, et d'autres informations de la base de données de Twitter. Cela en fait une ressource précieuse pour les chercheurs, les entreprises et les développeurs.

L'une des principales caractéristiques de l'API Twitter est sa capacité à fournir un accès aux données en temps réel. Les développeurs peuvent récupérer les tweets au fur et à mesure de leur publication, ce qui leur permet d'analyser et de traiter les dernières informations en temps quasi réel. C'est très utile pour surveiller les événements, suivre les tendances et mener des recherches sensibles au temps.

L'API de Twitter propose également des fonctionnalités de filtrage et de recherche. Les développeurs peuvent récupérer des tweets en fonction de critères spécifiques tels que des mots-clés, des hashtags, la langue, la géolocalisation, les profils d'utilisateurs, ou des plages de temps spécifiques. Cela leur permet d'extraire les données pertinentes et de personnaliser le comportement de l'API en fonction de leurs besoins.

Les tweets obtenus via l'API contiennent des données textuelles riches qui peuvent être utilisées pour l'analyse. Chaque tweet comprend généralement le texte du tweet lui-même,

le nom d'utilisateur, l'heure à laquelle il a été publié, le nombre de retweets, le nombre de favoris, et d'autres métadonnées. Les tweets peuvent également contenir des hashtags, des mentions, des URL, et des contenus multimédias tels que des images ou des vidéos. Cette combinaison de données structurées et non structurées fournit des informations précieuses aux chercheurs et aux développeurs pour comprendre le comportement des utilisateurs, les sentiments exprimés, et pour catégoriser le contenu.

En résumé, l'API de Twitter est une porte d'entrée vers une vaste quantité de données en temps réel des médias sociaux. Ses fonctionnalités et sa flexibilité permettent aux développeurs d'accéder, d'analyser, et d'utiliser les informations riches disponibles sur Twitter. Cela ouvre de nombreuses possibilités d'applications dans des domaines tels que la science des données, la recherche sociale, le marketing et l'engagement des clients.

1.2 Caractéristiques générales des tweets

Les tweets sont des textes particuliers qui possèdent des caractéristiques uniques et précieuses pour l'analyse. Tout d'abord, ils sont courts et concis, limités à 280 caractères maximum, ce qui pousse les utilisateurs à être brefs dans leurs messages. Cette brièveté représente un défi et une opportunité pour les chercheurs et les développeurs, car ils doivent extraire des informations importantes d'un texte limité. De plus, les tweets utilisent souvent un langage informel, avec des mots familiers, des abréviations et des émoticônes, reflétant le côté décontracté et conversationnel de la plateforme. Comprendre et traiter ce langage informel est essentiel pour analyser les sentiments, modéliser le langage et effectuer des tâches de traitement du langage naturel.

Une autre caractéristique des tweets est l'utilisation fréquente des hashtags. Les hashtags sont des mots-clés ou des phrases précédés du symbole "#", qui servent à classer et organiser les tweets autour de sujets spécifiques. Les hashtags facilitent la découverte de contenu, le suivi des tendances et l'analyse thématique. En étudiant l'utilisation et la popularité des hashtags, les chercheurs et les entreprises peuvent se faire une idée des tendances émergentes, du sentiment du public et des intérêts des utilisateurs.

Les mentions et les réponses sont également importantes dans les tweets. Les utilisateurs peuvent mentionner d'autres comptes Twitter en incluant leur nom d'utilisateur (précédé de "@") dans leurs tweets. Cela permet d'avoir des conversations, des interactions et de former des réseaux entre les utilisateurs. L'analyse des mentions et des réponses permet de comprendre l'engagement des utilisateurs, leur influence et la dynamique des communautés en ligne.

De plus, les tweets contiennent souvent des URL, des pièces jointes multimédias et des contenus multimédias. Les URL partagées dans les tweets peuvent fournir un contexte supplémentaire ou diriger les utilisateurs vers des ressources externes. Les pièces jointes multimédias telles que les images, les vidéos ou les GIF renforcent l'aspect visuel et expressif des tweets. L'intégration de ces éléments dans l'analyse permet de mieux comprendre l'engagement des utilisateurs, la viralité du contenu et l'expression des sentiments.

Enfin, les *Times-tamps* associés aux tweets fournissent des informations temporelles qui permettent d'analyser les tendances, de détecter les événements et d'étudier le comportement des utilisateurs au fil du temps. Les chercheurs peuvent examiner les schémas

d'activité des tweets, identifier les périodes d'engagement maximal et étudier la manière dont l'information se propage sur la plateforme.

En conclusion, les tweets en tant que données textuelles sont une source d'informations riche et dynamique qui peut être exploitée pour l'analyse des sentiments, la modélisation des sujets, l'analyse du comportement des utilisateurs et d'autres applications dans le domaine du TALN et de la recherche sur les médias sociaux.

1.3 Préparation des données

Avant d'analyser les tweets, il est important de préparer et de nettoyer les données pour obtenir des résultats fiables et précis. Ce processus comprend plusieurs étapes adaptées aux caractéristiques uniques des tweets en tant que données textuelles.

Tout d'abord, nous supprimons les tweets en double ainsi que les champs vides de notre base de données. Ensuite, nous retirons les émoticônes couramment utilisées dans les tweets, ainsi que les caractères spéciaux utilisés dans les hashtags et les mentions (les symboles "@" et "#"). Nous éliminons également les URL qui pourraient biaiser nos modèles.

Toutes ces étapes sont intégrées dans un pipeline, qui est une fonction Python utilisant la bibliothèque SpaCy (voir l'annexe 1) pour traiter et préparer ce type de données textuelles.

1.4 Conclusion

En conclusion, Twitter avec son API qui est un outil puissant pour accéder à une vaste quantité de données. Les tweets récupérés via cette API offrent des informations riches et uniques, ce qui en fait une ressource précieuse pour l'analyse des médias sociaux. Comprendre les caractéristiques spécifiques des tweets, telles que leur brièveté, leur langage informel, l'utilisation de hashtags et de mentions, ainsi que la présence d'URL et de contenus multimédias, est essentiel pour exploiter pleinement ces informations. La préparation minutieuse des données avant l'analyse garantit des résultats fiables.

2 Données CrisisNLP HumAID

L'ensemble de données CrisisNLP HumAID Twitter [**humaid2020**], est une collection complète de plusieurs milliers de tweets annotés manuellement, recueillis lors de 19 catastrophes naturelles majeures entre 2016 et 2019 dans différentes régions du monde. Ces événements comprennent des tremblements de terre, des ouragans, des incendies de forêt et des inondations. L'ensemble de données fournit des informations précieuses sur les aspects humanitaires de ces catastrophes grâce à ses annotations. Les annotations couvrent différentes catégories humanitaires, offrant des informations détaillées et une classification des tweets et des événements catastrophiques correspondants. Il est important de noter que l'ensemble de données HumAID ne contient que des tweets en anglais, ce qui assure la

cohérence de la langue utilisée dans l'ensemble des données. L'ensemble de données HumAID revêt une importance particulière en raison de sa taille et de sa couverture. Avec son grand nombre de tweets annotés manuellement, il s'agit de la collection de données la plus complète disponible à ce jour .

Les données Twitter HumAID constitue une ressource objective et instructive. En incluant des milliers de tweets annotés manuellement lors de catastrophes naturelles majeures et en mettant l'accent sur les catégories humanitaires, il est d'une valeur inestimable pour les chercheurs et les professionnels de l'informatique de crise. En exploitant cet ensemble de données, les chercheurs peuvent obtenir une compréhension approfondie de la dynamique des catastrophes et améliorer leurs connaissances sur la réponse aux crises et les efforts de rétablissement.

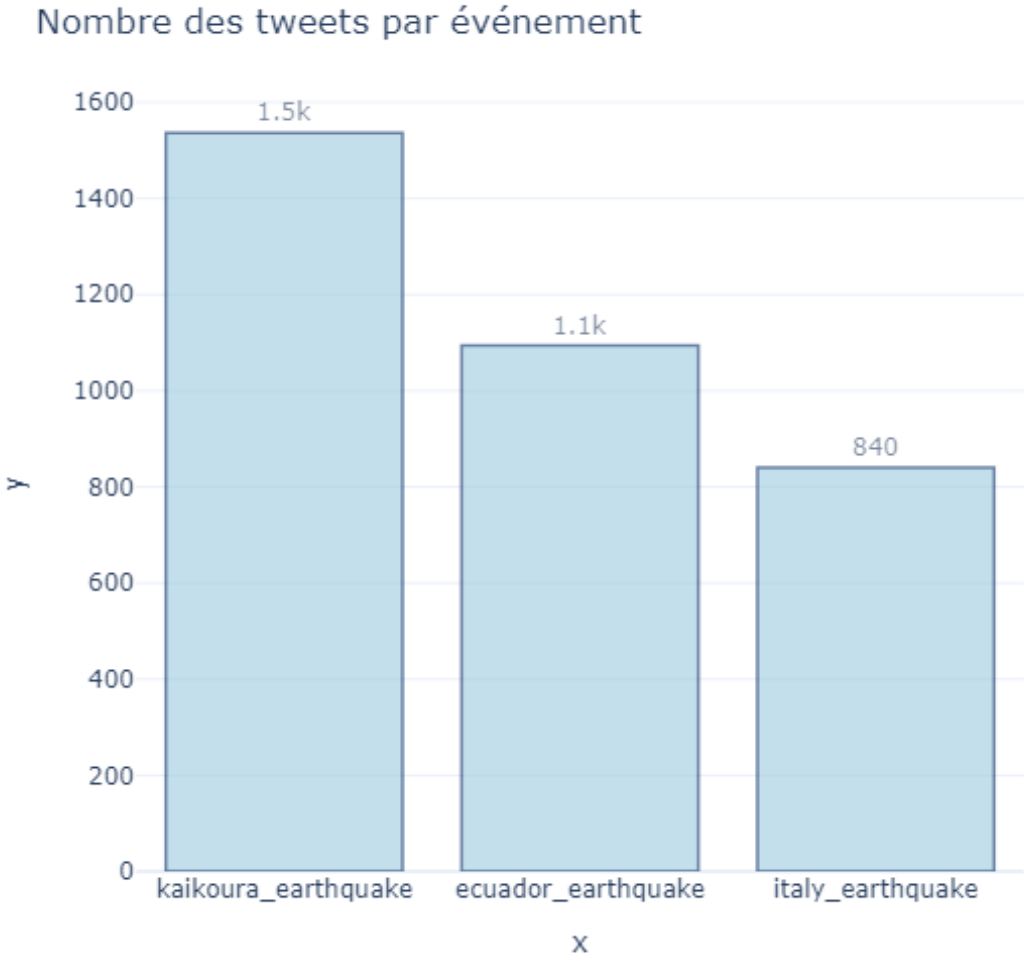

Fig. 3.2 : Nombre des tweets par événement sur les données HumanAID

Dans notre cas, nous nous concentrons uniquement sur trois événements de type séisme qui se sont produits la même année en 2016. Les tweets pour chaque événement sont répartis comme indiqué dans la figure 3.2. Le séisme de Kaikoura est l'événement avec le plus grand nombre de tweets, qui s'est produit en Nouvelle-Zélande. L'ampleur du séisme et sa portée expliquent la quantité de tweets étiquetés disponibles dans l'ensemble

de données. Ensuite, nous avons les deux autres événements, ceux de l'Équateur et de l'Italie, avec respectivement 1094 et 840 tweets étiquetés.

Les termes clés les plus fréquents dans les tweets concernant les trois séismes de Kaikoura, de l'Équateur et de l'Italie sont représentés respectivement dans les figures 3.3, 3.4 et 3.5. Nous remarquons que parmi les termes les plus fréquents pour le premier événement, nous avons "EQNZ" qui signifie "Earthquake New Zealand", pour le deuxième événement, nous avons "Ecuador", et pour le troisième événement, nous avons "PrayForItaly". Nous trouvons également d'autres termes qui sont liés à la géolocalisation ou aux effets de la catastrophe, tels que "death", "aftershock", "donate", etc. Ainsi, nous pouvons conclure que les tweets concernant ces trois événements contiennent des mots et des termes spécifiques à chaque crise, ce qui permet de distinguer facilement les tweets relatifs à des événements différents sans connaître leur étiquette.

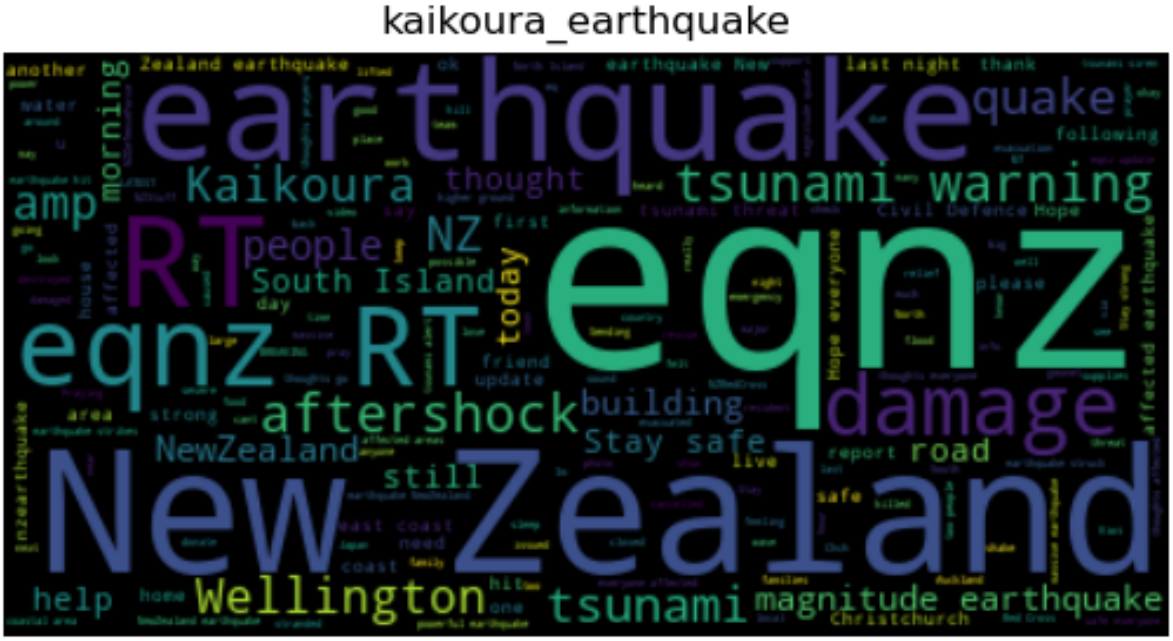

Fig. 3.3 : les mots très fréquent dans les tweets sur le séismes de Kaikoura (Word Cloud)

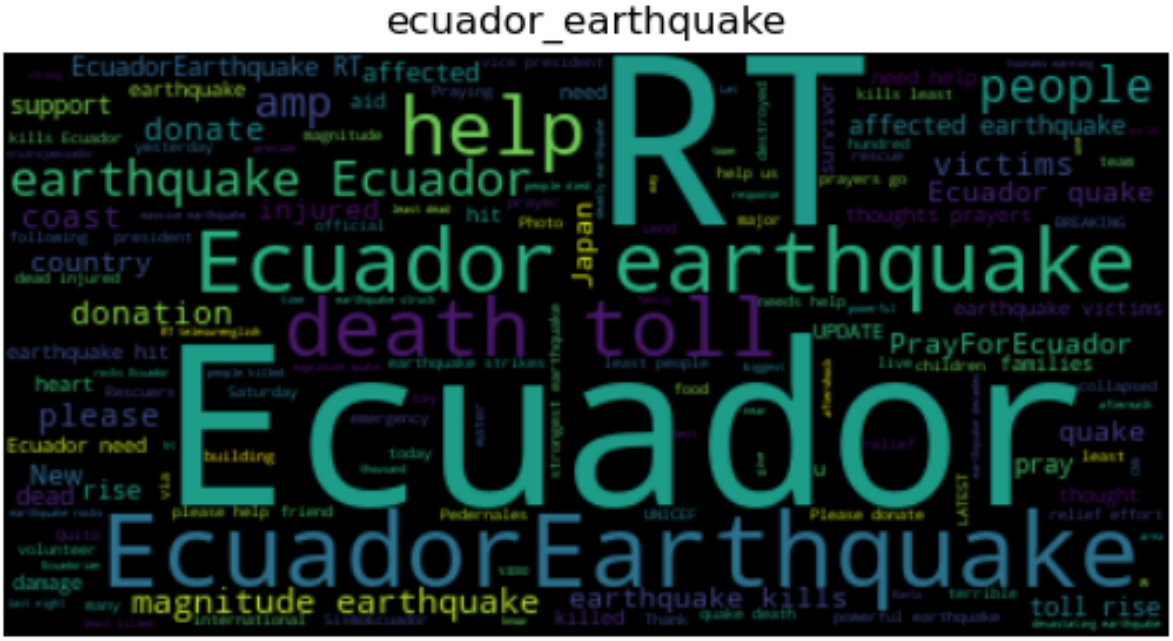

Fig. 3.4 : les mots très fréquent dans les tweets sur le séismes de l'Ecuador (Word Cloud)

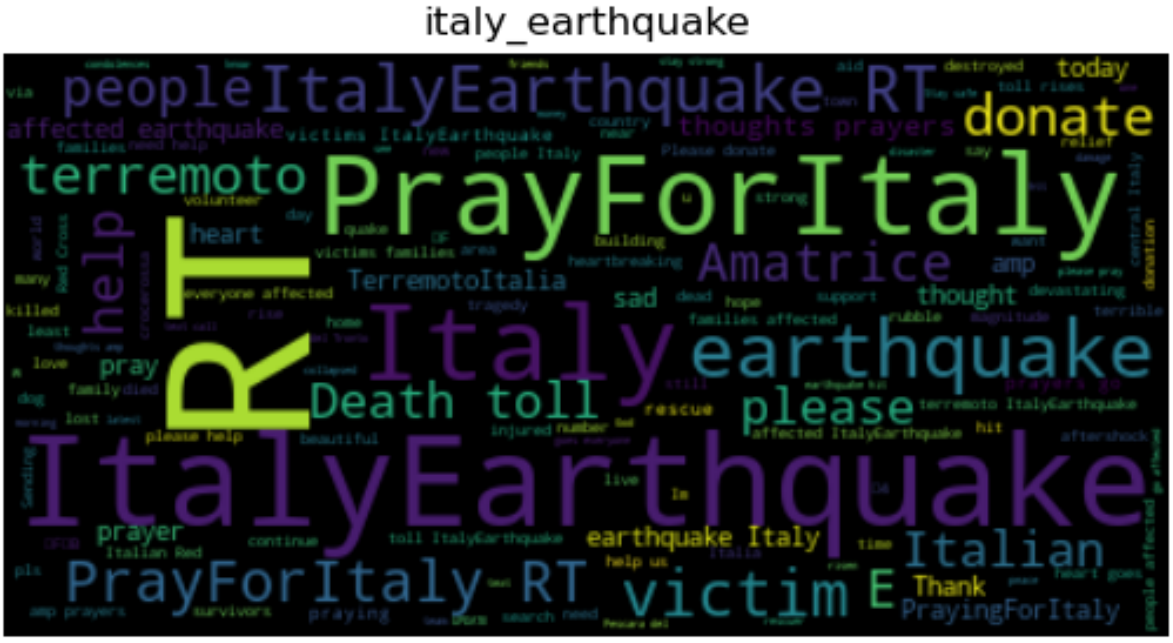

Fig. 3.5 : les mots très fréquent dans les tweets sur le séismes de l'italie (Word Cloud)

3 Données RéSoCIO

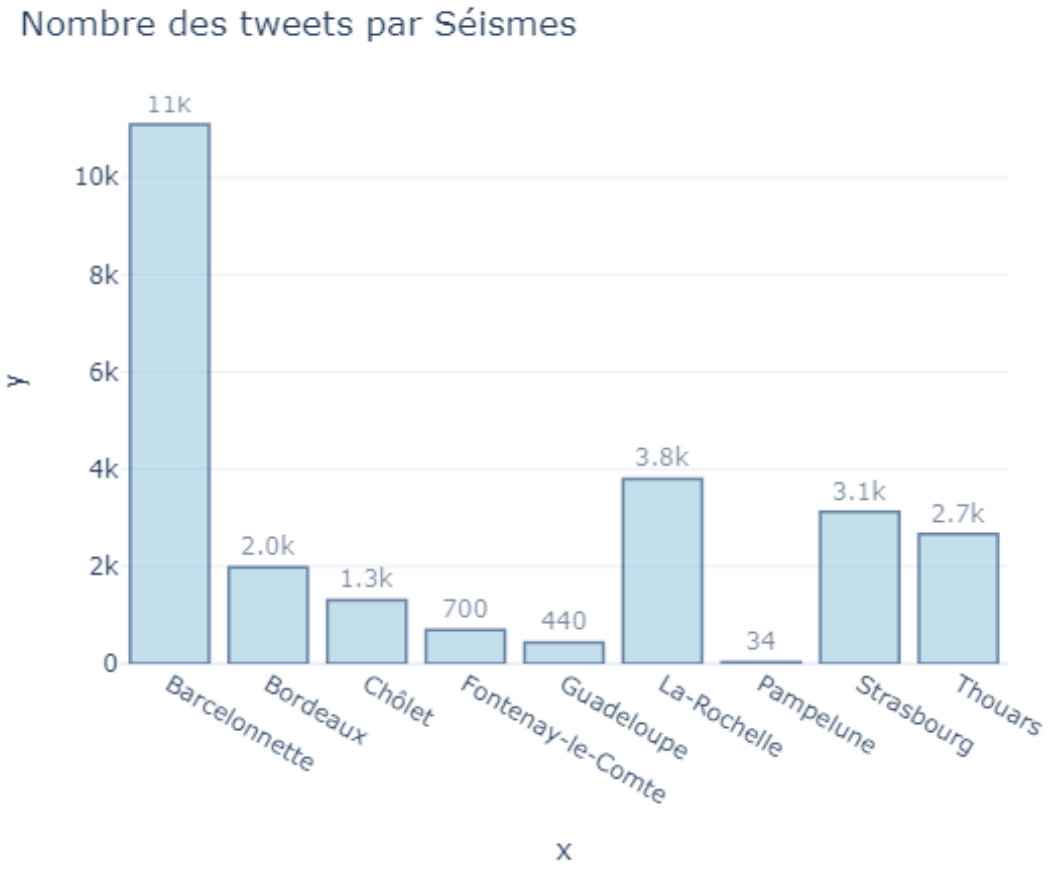

Fig. 3.6 : Nombre des tweets par événement de type séismes sur les données RéSoCIO

Les données RéSoCIO sont extraites de l'API de Twitter en utilisant des requêtes spécifiques pour collecter des tweets en français sur les séismes et les inondations, en excluant les retweets politiques. Les experts du BRGM utilisent des catalogues et établissent des règles pour chaque événement. En général, ils effectuent l'extraction en utilisant des requêtes sur l'API pour chaque événement, en délimitant l'intervalle de temps et en respectant les règles prédéfinies basées sur le catalogue de chaque événement. Il est important de noter que les données RéSoCIO ne sont pas annotées par des humains, contrairement à d'autres données utilisées dans ce projet.

Ces données RéSoCIO peuvent confirmer notre hypothèse initiale selon laquelle il n'est pas garanti d'avoir des tweets concernant un seul événement dans un laps de temps donné. La figure 3.7 présente le nombre de tweets émis au fil du temps pour chaque crise, révélant ainsi le chevauchement de différentes crises naturelles dans le même intervalle de temps.

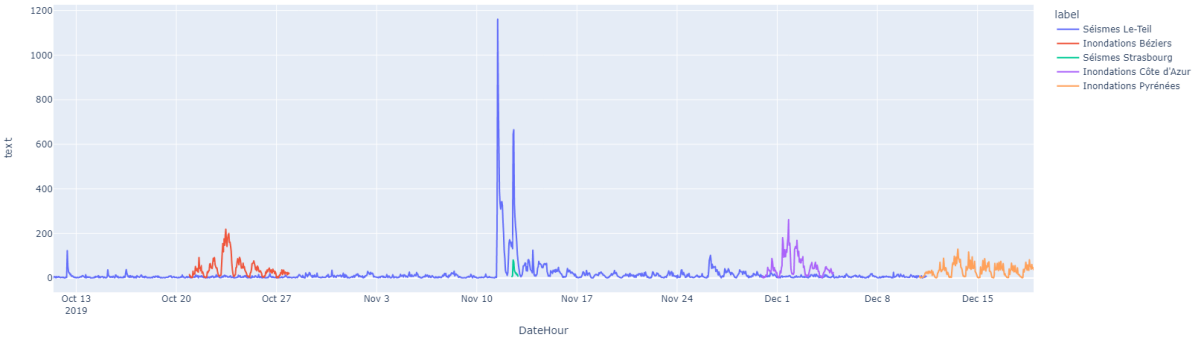

Fig. 3.7 : Cas de chevauchement des catstrophes naturelles dans le temps

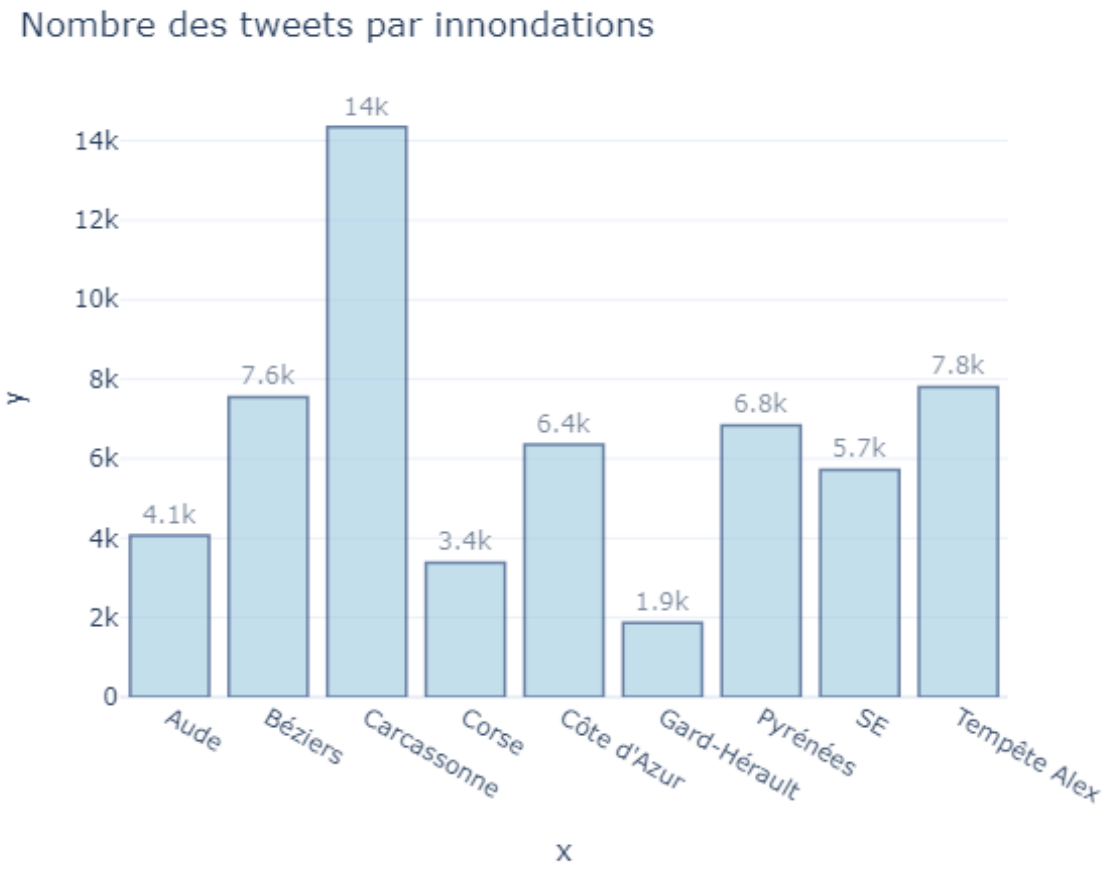

Fig. 3.8 : Nombre des tweets par événement de type inondations sur les données RéSoCIO

Pour les séismes, les tweets extraits incluent des mots tels que “seisme, seismes, séisme, séismes, tremblement de terre,tremblements de terre,magnitude et terre tremble”. Ces termes sont souvent utilisés pour discuter des tremblements de terre, de leur intensité et de leurs effets. Les tweets sélectionnés excluent tout contenu politique et se concentrent sur le contexte scientifiques et environnementaux des séismes.

En ce qui concerne les inondations, les tweets extraits incluent des mots-clés tels que “inondation, inondations,inondé, inondée, inondés, inondées, sous l’eau, en crue, la crue, crues, décrue, onde de crue, sort de son lit, sorti de son lit, torrentiel, emporté par les

eaux et emportée par les eaux". Ces termes permettent de recueillir la majorité des tweets contenant des informations sur les inondations, leur ampleur, les régions touchées et les dégâts causés. Les tweets sélectionnés excluent les sujets liés au sexe et à l'immigration, se concentrant plutôt sur le contexte des catastrophes naturelles.

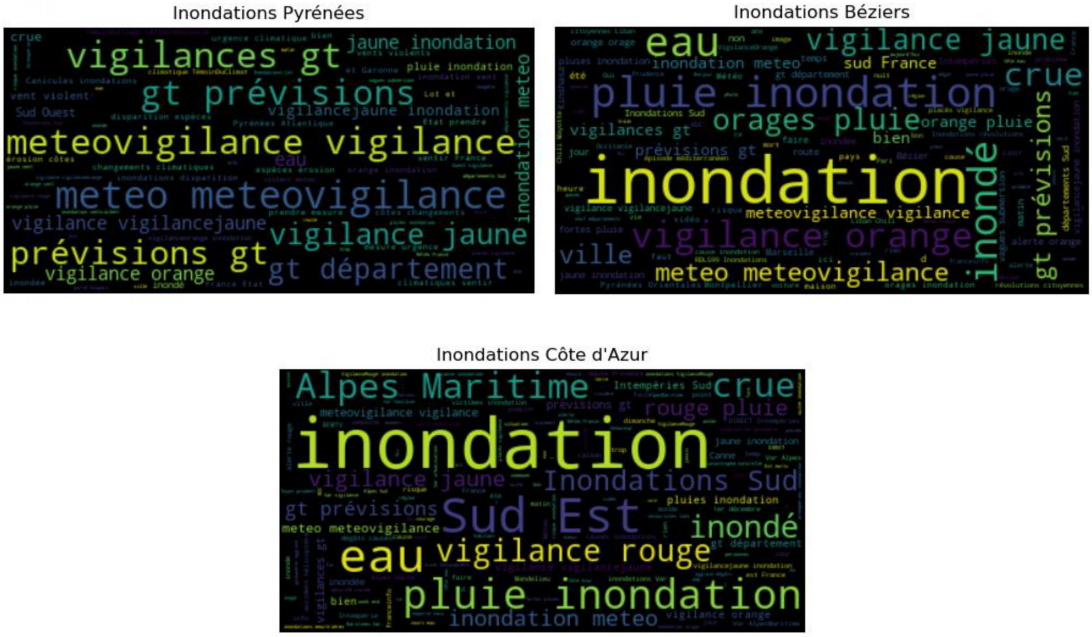

Fig. 3.9 : les mots très fréquent dans les tweets sur les inondations dans les données RéSoCIO

En utilisant ces requêtes spécifiques, nous avons obtenu un grand nombre de tweets, comme le montrent les deux figures 3.6 et 3.8. Pour les séismes et les inondations, nous observons des quantités de tweets différentes pour chaque événement, et cette quantité dépend de l'ampleur et de la zone géographique. Certains événements sont répartis sur une région plus large, tandis que d'autres se concentrent sur une seule ville, ce qui explique la répartition des tweets par événement.

En examinant de plus près le contenu des données RéSoCIO, les figures 3.9 et 3.10 montrent les termes les plus fréquents dans l'ensemble des tweets concernant les événements inclus dans les données RéSoCIO. On remarque que ces termes fréquents sont très généraux et ressemblent davantage aux termes utilisés dans les requêtes d'extraction. Contrairement aux autres données que nous utiliserons ultérieurement, la plupart des tweets extraits associés à ces événements ne contiennent pas d'informations spécifiques ni de termes clés pertinents.

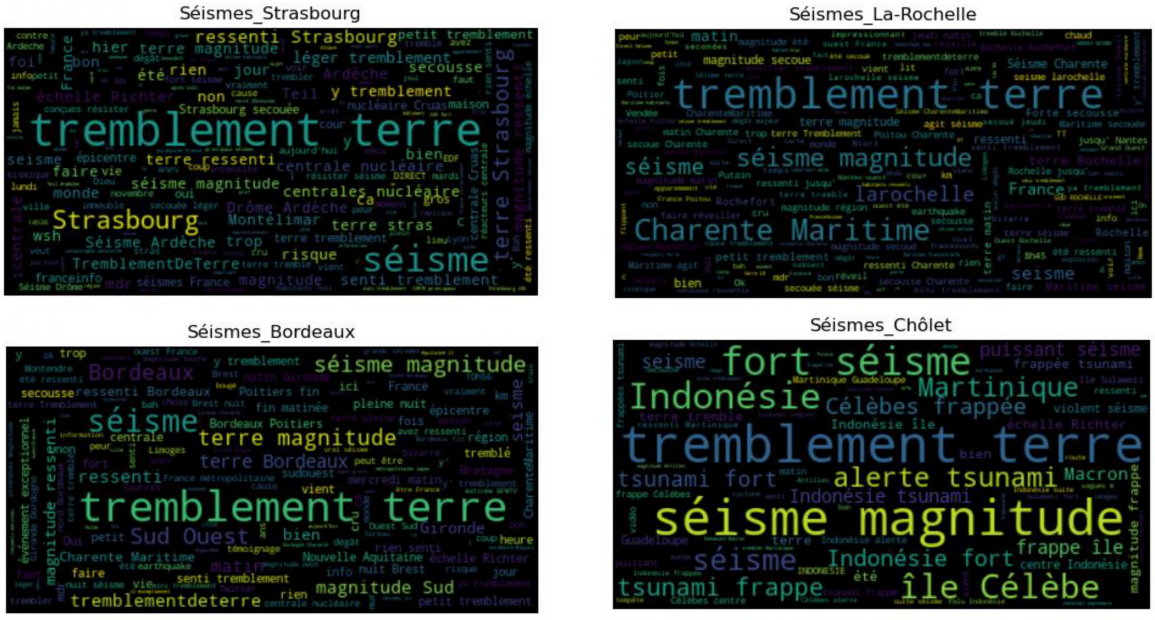

FIG. 3.10 : les mots très fréquent dans les tweets sur les séismes dans les données RéSoCIO

4 Données filtrées

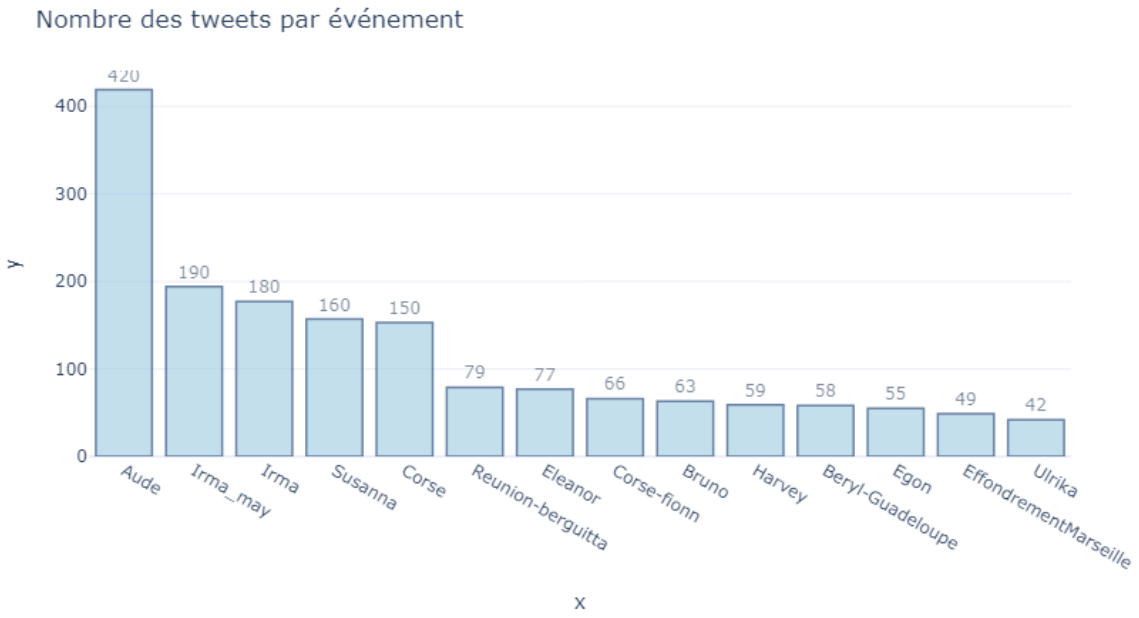

FIG. 3.11 : Nombre des tweets par événement sur les données filtré

En 2020, des chercheurs ont publié l'article intitulé "Une classification à trois niveaux

des tweets français sur les crises écologiques" [9]. Dans leur travail, ils ont créé un jeu de données comprenant des tweets en français annotés manuellement selon trois dimensions : la pertinence, les intentions d'action et l'urgence. Ils ont affiné et adapté le concept de pertinence situationnelle afin de répondre aux exigences du Département français de la Sécurité Civile et de la Gestion des Crises.

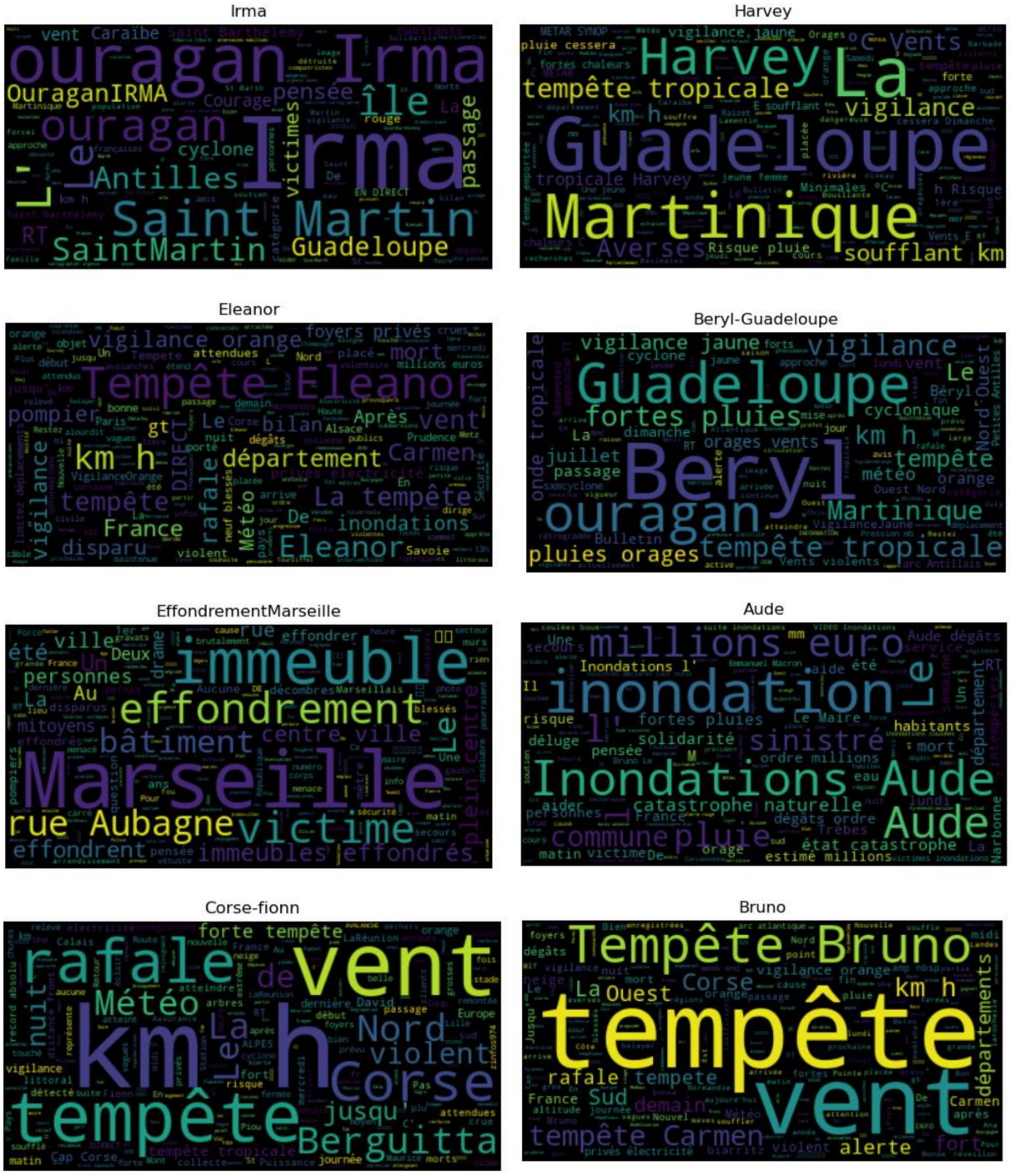

Fig. 3.12 : les mots très fréquent dans les tweets sur les événements des données filtré

L'ensemble de données utilise une taxonomie à trois niveaux pour classer les tweets en fonction de leur utilité, de leur caractère urgent (non utile, utile et urgent, non urgent) et des intentions d'action spécifiques. Cette catégorisation a été conçue pour répondre aux

besoins des acteurs sur le terrain impliqués dans la gestion des crises. Le but est de fournir des informations précises et ciblées aux personnes chargées de prendre des décisions et d'agir en cas de crise écologique.

Ce travail est important pour notre projet. Bien que le nombre de tweets annotés soit relativement petit (figure 3.11) par rapport à ceux que nous avons mentionnés précédemment, ils contiennent de termes et d'informations pertinentes associés à chaque événement, comme le montre la figure 3.12. On remarque la présence de termes fréquents qui facilitent la distinction entre les différentes classes. Ces tweets annotés fournissent donc des informations précieuses pour chaque événement.

5 Synthèse

Ce chapitre se concentre sur l'exploration et la compréhension des données recueillies à partir de l'API de Twitter et d'autres outils. Nous avons examiné plusieurs jeux de données, notamment CrisisNLP HumAID, RéSoCIO et un ensemble de données filtrées.

L'ensemble de données CrisisNLP HumAID est une collection riche de milliers de tweets annotés manuellement, collectés lors de 19 catastrophes naturelles majeures. En se concentrant sur des aspects humanitaires spécifiques, il offre une valeur inestimable pour l'analyse des réponses aux catastrophes. Nous avons examiné des tweets liés à trois tremblements de terre majeurs et observé que chaque événement avait des termes clés et spécifiques qui peuvent aider à les distinguer.

Les données RéSoCIO, quant à elles, ont été recueillies par l'intermédiaire de l'API de Twitter à l'aide de requêtes spécifiques. Ces données ne sont pas annotées par des humains mais fournissent des informations précieuses sur la dynamique des discussions sur les séismes et les inondations sur Twitter en français. Nous avons remarqué que les termes les plus fréquemment utilisés dans ces discussions sont généraux et correspondent aux termes utilisés pour l'extraction.

Enfin, nous avons également examiné un ensemble de données filtrées, créé par des chercheurs et annoté manuellement. Bien que l'ensemble de données soit plus petit, il offre des informations précieuses car il est spécifiquement adapté pour identifier la pertinence, l'urgence et les intentions d'action.

En conclusion, il est certain que Twitter présente un potentiel considérable pour l'étude et la compréhension des catastrophes naturelles et des réponses à ces événements. Chaque jeu de données possède ses propres avantages et caractéristiques spécifiques. Par conséquent, nous utiliserons ces trois bases de données pour expérimenter notre modèle dans le chapitre 5. Dans l'autre chapitre 4, notre attention sera uniquement portée sur la dernière base de données filtrée. Il est important de souligner que pour notre problématique, il n'existe pas de données de référence ou de benchmark. Nous avons donc choisi ces bases de données qui nous semblent être les plus adaptées à notre problème.

Chapitre IV

Approche basée sur le clustering

Dans le prochain chapitre, nous présenterons le premier travail réalisé après avoir étudié les travaux existants. Après une analyse approfondie des différentes approches, il est clair que la méthode SCCL, basée sur le clustering, se présente comme une solution particulièrement adaptée à notre problématique. Comme nous l'avons souligné dans la conclusion de notre revue de la littérature, la méthode SCCL offre des avantages significatifs pour le regroupement de textes, en particulier lorsqu'elle est appliquée à des tweets.

En effet, la méthode SCCL utilise des techniques innovantes qui améliorent considérablement la qualité du regroupement des textes. De plus, l'un de ses atouts majeurs est qu'elle ne nécessite pas de grandes quantités de données, grâce à l'utilisation d'une technique d'augmentation de texte. Ces caractéristiques ont suscité notre intérêt et nous ont convaincus de mettre en œuvre la méthode SCCL pour notre problème qui vise à regrouper les tweets qui parlent de la même crise. Ainsi, ce chapitre commencera par une introduction où nous définirons les techniques et les principes clés nécessaires à une bonne compréhension de l'architecture du modèle SCCL. En mettant en évidence les motivations tirées de notre revue de la littérature, nous établirons une solide base pour la présentation de cette méthode. Ensuite, nous détaillerons l'architecture du modèle SCCL, en expliquant ses composants essentiels et leur fonctionnement. Enfin, nous présenterons les expérimentations que nous avons réalisées dans ce domaine, afin de mettre en lumière les performances et l'efficacité de la méthode SCCL dans le contexte de notre problématique.

1 Introduction

L'introduction du framework SCCL en 2021 dans la publication "Supporting clustering with contrastive learning" [23] représente une approche novatrice pour soutenir le clustering non supervisé. En intégrant l'apprentissage contrasté au clustering non supervisé, SCCL améliore efficacement la séparation entre les catégories dans l'espace de représentation. Les résultats expérimentaux, notamment dans le domaine du clustering de courts textes, démontrent la capacité de SCCL à améliorer significativement les performances de regroupement, permettant ainsi une meilleure identification des catégories et des résultats de clustering plus fiables.

Le clustering non supervisé est le processus de découverte de catégories sémantiques ou de groupes au sein d'un ensemble de données en se basant sur les distances entre les points de données dans un espace de représentation. Cependant, aux premiers stades de l'apprentissage, différentes catégories peuvent se chevaucher dans cet espace, ce qui rend difficile pour les algorithmes de clustering basés sur les distances d'obtenir une bonne séparation entre les catégories.

Pour relever ce défi, les auteurs proposent SCCL, qui exploite l'apprentissage contrastif pour favoriser une meilleure séparation entre les groupes des données. SCCL vise à améliorer la séparation entre les catégories en appliquant l'apprentissage contrastif dans le contexte du clustering.

1.1 L'apprentissage contrastif

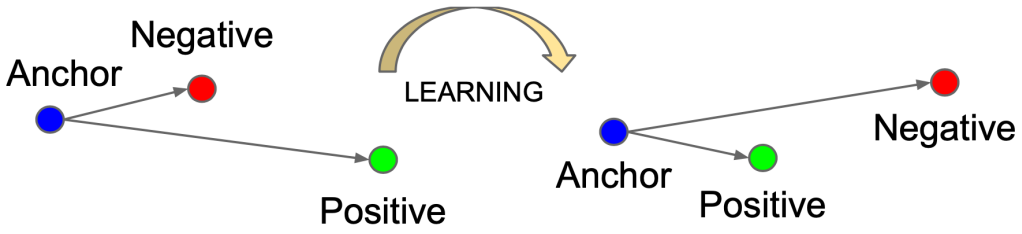

Fig. 4.1 : Démonstration du processus clés de l'apprentissage contrastif

L'apprentissage contrastif est une technique dans laquelle nous entraînons un modèle à différencier les paires de données similaires et différentes en maximisant leur similarité au sein de la même classe et en la minimisant entre différentes classes.

Cette technique a de nombreuses applications, notamment en vision par ordinateur et en traitement du langage naturel. L'un des principaux avantages de l'apprentissage contrastif est sa capacité à apprendre à partir de données non étiquetées, ce qui le rend très évolutif
.

Cependant, cette technique présente plusieurs défis, notamment la recherche d'exemples négatifs appropriés, la sélection de la bonne taille de lot et la minimisation des faux négatifs. Nous allons explorer le concept de l'apprentissage contrastif, ses cas d'utilisation, la fonction du *Contrastive loss* et certaines considérations à prendre en compte lors de sa

mise en œuvre.

L'idée derrière l'apprentissage contrastif est que nous avons un échantillon de référence ou "ancre", un échantillon similaire ou "positif" et un échantillon différent ou "négatif" (voir figure 4.1). Nous essayons de rapprocher les échantillons positifs de l'échantillon ancre dans un espace d'incorporation tout en éloignant les échantillons négatifs.

Dans l'exemple de TALN, nous pourrions essayer d'apprendre un encodeur (par exemple, un Transformer) qui rapproche les représentation des données textuelles positives et éloigne les représentations négatives. Un échantillon positif pourrait être un texte de la même classe que l'ancre ou une version augmentée du texte ancre, tandis que l'échantillon négatif serait un texte totalement différente (généralement d'une classe différente).

Dans sa forme générale, la formule du *Contrastive loss* mesure la similarité entre deux exemples en utilisant une fonction de distance. La formule de base est souvent exprimée comme suit :

$$\boxed{\text{Loss} = \frac{1}{2N}\sum_{i=1}^{N}(Y \cdot D^2 + (1-Y)\cdot \max(\text{margin} - D, 0)^2)} \tag{IV.1}$$

où :

- N : nombre total de paires d'exemples dans le lot (batch).
- Y : variable binaire indiquant si les exemples sont de la même classe (1) ou de classes différentes (0).
- D : distance entre les deux exemples.
- margin : hyperparamètre contrôlant la marge minimale souhaitée entre les exemples de classes différentes.

1.2 L'augmentation des données textuelles

L'augmentation des données textuelles est une technique essentielle dans le traitement du langage naturel (TALN). Elle permet d'améliorer la précision et la robustesse des modèles d'apprentissage automatique en augmentant la quantité et la diversité des données d'entraînement. En d'autres termes, l'augmentation des données textuelles peut aider à améliorer les performances des modèles TALN en les rendant plus efficaces et performants.

Cette technique consiste à générer de nouveaux échantillons de texte à partir de ceux existants, ce qui permet de créer un ensemble de données plus volumineux pour l'entraînement du modèle. Cela résout le problème d'un ensemble de données d'entraînement limité, qui entraîne souvent un overfitting ou de mauvaises performances de généralisation. En ajoutant des exemples supplémentaires représentant différentes variations, tels que des synonymes, des paraphrases et des variations de structures grammaticales, la diversité du corpus augmente.

L'augmentation des données est bénéfique dans des tâches de TALN telles que l'analyse des sentiments, les systèmes de questions-réponses et la traduction, notamment lorsque

peu d'ensembles de données étiquetées sont disponibles. Ces techniques permettent d'augmenter la quantité de données étiquetées sans avoir à effectuer un processus d'étiquetage manuel fastidieux, qui prend du temps.

Il existe différentes techniques d'augmentation des données. Parmi celles-ci, on peut trouver des techniques basées sur le remplacement par des synonymes. Dans ce sens, on peut citer deux techniques très utilisées :

WordNet Augmenter [15, 18] qui est une technique d'augmentation qui modifie un texte en remplaçant ses mots par des synonymes provenant de WordNet. WordNet est une base de données lexicale qui regroupe les mots en synsets, qui sont des ensembles de synonymes représentant différentes significations d'un mot. Cette technique utilise la vaste collection de synonymes de WordNet pour suggérer des mots alternatifs pouvant remplacer les mots d'origine dans le texte donné. Il exploite les relations sémantiques capturées par WordNet, telles que les synonymes, les hyperonymes (termes plus généraux), les hyponymes (termes plus spécifiques) et les méronymes (relations de parties et de tout), pour trouver des remplacements appropriés.Lors de l'application du "WordNet Augmenter", chaque mot du texte est pris en compte et ses synonymes provenant de WordNet sont récupérés. L'Augmenter sélectionne un synonyme en fonction de sa pertinence et de son adéquation dans le contexte donné. Le synonyme choisi est ensuite substitué au mot original dans le texte de sortie.

Contextual Augmenter [8] qui est une technique d'augmentation qui utilise des *Transformers* pré-entraînés pour identifier et proposer des mots appropriés à insérer ou à substituer dans un texte donné. Elle exploite la compréhension contextuelle et les connaissances sémantiques capturées par ces modèles de *Transformers* afin de générer des suggestions pertinentes et adaptées au contexte. Le processus consiste à soumettre le texte d'entrée au modèle de *Transformer* pré-entraîné, qui encode le texte et en capture la représentation sémantique. En analysant les informations contextuelles, le modèle identifie les mots ou les phrases qui peuvent être insérés ou substitués pour améliorer le texte. Le terme "top-n" fait référence au nombre de suggestions ou de mots candidats générés par l'augmentation contextuelle pour chaque opération d'insertion ou de substitution. Le modèle classe ces suggestions en fonction de leur pertinence et de leur adéquation par rapport au contexte donné.

Dans la suite de ce chapitre, nous utiliserons le *Contextual Augmenter*, qui a démontré de meilleures performances dans les tests de la publication originale du SCCL [23]. Cette technique d'augmentation des données textuelles exploite les *Transformers* pré-entraînés pour proposer des mots appropriés à insérer ou à substituer dans un texte donné. En exploitant la compréhension contextuelle et les connaissances sémantiques des modèles de *Transformers*, le *Contextual Augmenter* génère des suggestions pertinentes et adaptées au contexte. Cette approche se révèle particulièrement efficace pour améliorer les performances du modèle SCCL, en augmentant la quantité des données d'entraînement.

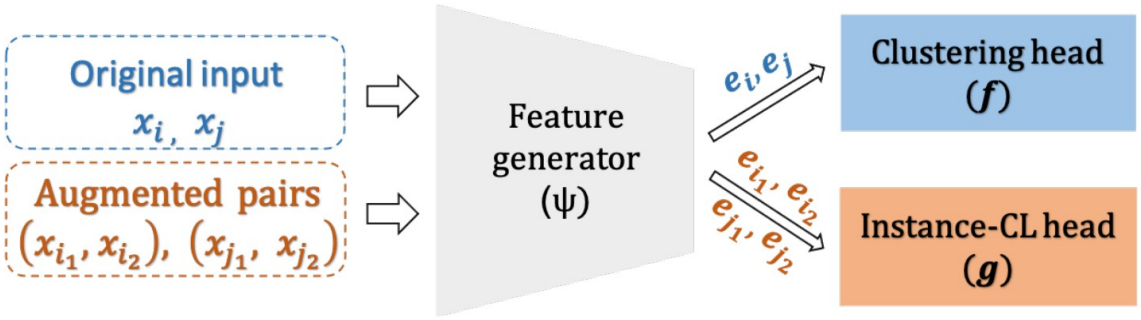

FIG. 4.2 : Framework d'entrainement SCCL [23]

2 Architecture du modèle

Le modèle SCCL comprend trois composants principaux (voir la figure 4.2) : un réseau neuronal de base appelé *Feature generator*, et deux autres composants, le *Instance-CL head* et le *Clustering head*. Le *Feature generator* est responsable de la génération des représentations des données d'entrée. Une fois ces représentations générées, le *Contrastive loss* est calculé entre la représentation des paires positive et les négatives, sachant que les paires positives sont deux augmentations différentes de la même instance, et les paires négatives sont des instances de différentes données originales. En minimisant ce *loss* , le modèle force le *Feature generator* à générer des représentations similaires pour minimiser les distances entre les différentes instances de la même catégorie et l'éloigner entre les autres. Sur la base de ces représentations, un algorithme de regroupement est appliqué pour déterminer si un cluster regroupe les instances de la même catégorie ou non. Cela se fait en calculant le *Clustering loss* dans le composant *Clustering head*. C'est là que l'on voit la contribution de l'apprentissage contrastif (*Contrastive learning*) dans l'amélioration du processus de regroupement du texte. Pour détailler les deux composants, on peut les décrire comme suit :

Pour chaque mini-batch $B = x_{i i=1}^{M}$,tels que x_i, est le tweet originale et M le nombre des tweets dans chaque mini-bath , on génére deux augmentations pour chaque instance x_i,en créant un lot augmenté B_a de taille $2M$.

Le composant Instance-CL Dans ce composant chaque instance de B_a est prise, et pour chaque instance, une paire positive est choisie, tandis que les autres 2M-2 exemples dans B_a sont considérés comme des instances négatives. Le modèle cherche à séparer la paire positive des instances négatives. Cela se fait à l'aide du *Contrastive loss* qui est calculée sur la base d'une mesure de similarité entre les représentations des instances. La mesure de similarité est un produit scalaire entre les sorties normalisées. le *Contrastive loss* est ensuite moyennée sur toutes les instances de B_a.

Le composant Clustering head C'est la deuxième partie principale de l'architecture du modèle qui concerne le Clustering. Le modèle intègre un composant de regroupement non supervisé qui tente de coder les concepts et catégories sémantiques de haut niveau dans les données. Zhang et al. [23] suggère que le composant de clustering cherche à regrouper des instances de la même catégorie sémantique. Les données comprennent K catégories

sémantiques, chacune caractérisée par son centroïde μ_k dans l'espace de représentation. Le modèle utilise alors la distribution de Student pour calculer la probabilité d'assigner une instance x_j au k-ième cluster. L'objectif donc est d'optimiser la divergence de Kullback-Leibler (KL divergence) entre la distribution cible et la distribution d'assignation de groupes, afin d'aligner les assignations avec la structure de regroupement souhaitée.

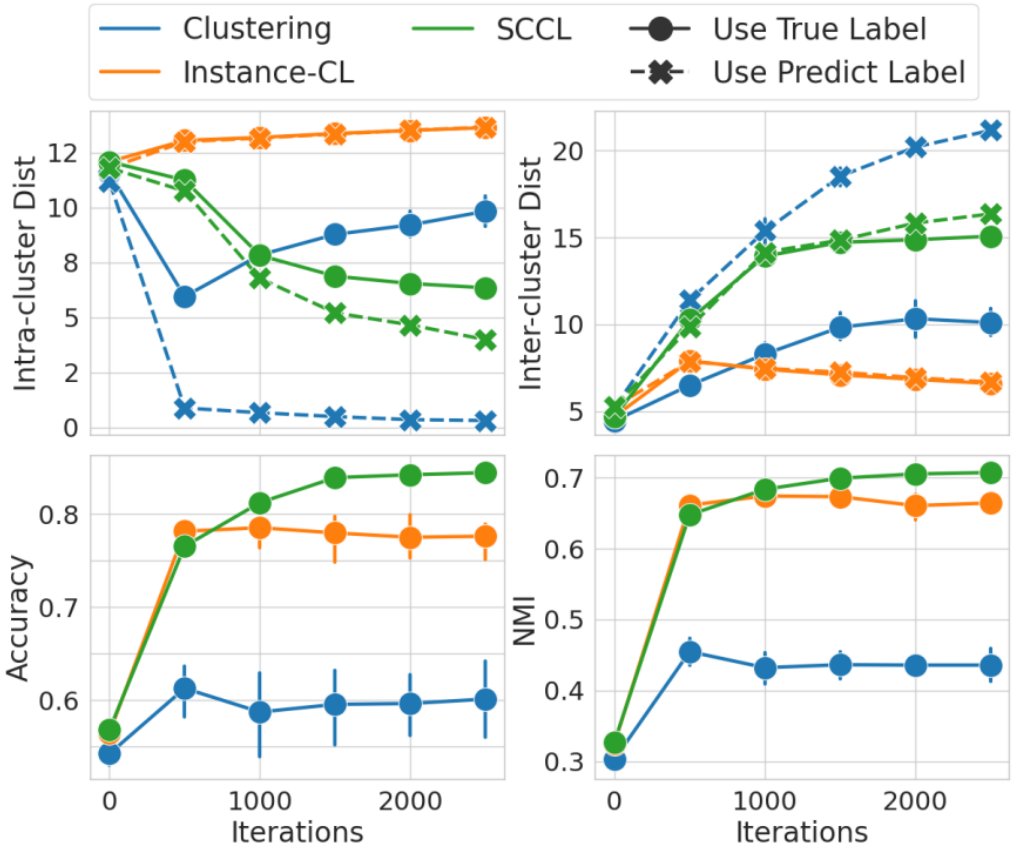

FIG. 4.3 : Evaluation sur SearchSnipets au niveau des clusters [23]

Pour mieux comprendre pourquoi le modèle SCCL surpasse les autres modèles, les auteurs étudient les distances intra-cluster et inter-cluster pendant le processus d'apprentissage (voir la figure 4.3). La distance intra-cluster mesure la distance moyenne entre le centroïde d'un cluster et tous les échantillons à l'intérieur de ce cluster. La distance inter-cluster est la distance entre un cluster et son cluster voisin le plus proche.

D'après les résultats dans la figure 4.2, on observe que la technique de Clustering obtient des distances intra-cluster plus petites et des distances inter-cluster plus grandes lorsqu'elle est évaluée sur des clusters prédits. Cela indique l'efficacité du Clustering à regrouper étroitement des clusters auto-apprentis et à séparer différents clusters. Cependant, lorsqu'il a été évalué sur les clusters de vérité de terrain, l'effet inverse a été observé, indiquant un chevauchement potentiel dans l'espace d'intégration avant le début du clustering.

En revanche, l'Instance-CL a obtenu de meilleurs scores de précision et de NMI grâce à son effet implicite au processus de regroupement. Cependant, cela a conduit à ce que les clusters soient moins séparés les uns des autres et plus dispersés dans l'espace d'intégration,

comme l'indiquent des distances inter-cluster plus petites et des distances intra-cluster plus grandes. Le modèle SCCL, en combinant les forces du Clustering et de l'Instance-CL, a abouti à des clusters mieux séparés, chaque cluster étant moins dispersé. Les résultats montrent que le SCCL améliore efficacement les avantages de l'Instance-CL et du Clustering, aboutissant à un modèle qui réalise à la fois une haute performance et un regroupement significatif des données.

3 Expérimentation et résultats

Le Framework SCCL a été testé sur diverses bases de données dans la littérature, y compris les tweets. Cependant, les chercheurs qui ont publier n'ont pas spécifié dans quels domaines les tweets ils en ont parlé. Lors de notre stage, nous avons tenté d'expérimenter le modèle SCCL avec les configurations définies et recommandées dans la publication originale [23], en modifiant quelques paramètres pour s'adapter aux caractéristiques de nos données.

Nous avons réalisé trois expériences. Celles-ci ont principalement consisté dans l'entrainement d'un modèle SCCL sur un cluster de calcul équipé d'un GPU. Durant l'expérimentation, nous avons formé le modèle sur des tweets relatifs à diverses crises naturelle, en considérant que chaque crise doit former un cluster de tweets particulier. Dans ces circonstances, notreobjectif était d'associer les représentations de chaque tweet relatif à une crise donnée à un cluster. Pour la stratégie d'entraînement et de test, nous avons suivi les mêmes protocoles que ceux définis dans le cadre SCCL. Ces protocoles consistent dans l'entraînement d'un modèle pour un nombre donné d'itérations et à voir la performance sur validation après chaque 100 itérations. Pour les données, nous avons utilisé 80% pour l'entraînement et 20% pour la validation. Ainsi, à chaque étape de validation, nous avons calculé les métriques suivantes :

- **NMI** : une métrique utilisée pour mesurer la similarité entre deux ensembles de clusters dans le contexte du clustering. Il se calcule en utilisant la matrice de contingence, qui représente le nombre de points communs entre chaque paire de clusters, ainsi que les distributions marginales des clusters dans chaque ensemble. Le NMI fournit une valeur comprise entre 0 et 1, où une valeur proche de 1 indique une forte similarité entre les ensembles de clusters, suggérant ainsi une bonne qualité de clustering.

- **Cluster accuracy** : est une mesure de précision spécifiquement conçue pour évaluer les performances de l'algorithme de clustering SCCL. Elle prend en entrée les étiquettes réelles et les étiquettes prédites des échantillons et utilise une matrice de confusion pour calculer la précision du clustering. La fonction attribue des valeurs à la matrice de confusion en comparant les étiquettes prédites et réelles, puis utilise l'algorithme de la méthode hongroise pour trouver l'affectation optimale des clusters. Enfin, elle calcule la précision en divisant la somme des valeurs maximales des paires (étiquette prédite, étiquette réelle) par la taille des étiquettes prédites.

- **Clustering loss** : c'est la loss calculée par le *loss* KCL (Kullback-Leibler Clustering) qui est une fonction de *loss* utilisée dans le contexte du clustering. Elle est

conçue pour mesurer la divergence de Kullback-Leibler (KLD) entre deux distributions de probabilité, généralement les distributions prédites et cibles. le *loss*KCL utilise une fonction KLDiv qui calcule la divergence de KLD entre les deux distributions et retourne la moyenne de ces divergences sur l'ensemble du lot. ce *loss* vise à guider l'entraînement de l'algorithme de clustering en minimisant la divergence de KLD entre les distributions, favorisant ainsi la prédiction de distributions similaires à la distribution cible. En résumé, le *loss*KCL est utilisée pour mesurer et optimiser la similarité entre les distributions de probabilité dans le clustering.

3.1 Expérimentation 1

Dans cette expérimentation, nous avons entraîné le modèle sur les données HumanAID (voir la section 1.4)qui comprend des tweets en anglais relatifs à trois catastrophes naturelles qui se sont produites en 2016. Nous avons entraîné et validé le modèle en utilisant les configurations et les paramètres indiqués (voir tableau 4.1). L'objectif initial de cette expérimentation était de tester le fonctionnement de cette méthode avec un corpus sur les catastrophes naturelles.

Paramètre	Valeur
Taille du batche	400
Nb des classes	3
Nb Itération	2000
Marge CL	0.5
Taux d'apprentissage	1e-05
Encodeur	BERT
Algorithme de clustering	KMeans
Temps d'exécution	5-6 hr

Tab. 4.1 : Configuration du SCCL de la première Expérimentation

Les résultats de cette expérimentation démontrent de bonnes performances du modèle sur les données de validation. Ces performances peuvent être observées dans les graphiques ci-dessous. Tout d'abord, nous remarquons la convergence et la minimisation de la fonction de *clustering loss* dans le graphique 4.4,ce qui permet de conclure que les clusters qui sont construit par le modèle sont très proches au termes de l'information à la distribution des tweets par classes, autrement dit qu'un cluster est plus proches d'un ensemble des instances qui parlent de la même crise. De plus, l'expérimentation donne un très bon NMI (Indice de Mutual Information Normalisé), avec une moyenne de 0,80, comme évalué lors des itérations dans le graphique 4.5. On peut aussi observer l'évolution de la précision du clustering dans le graphique 4.6, avec une moyenne de 0,94.

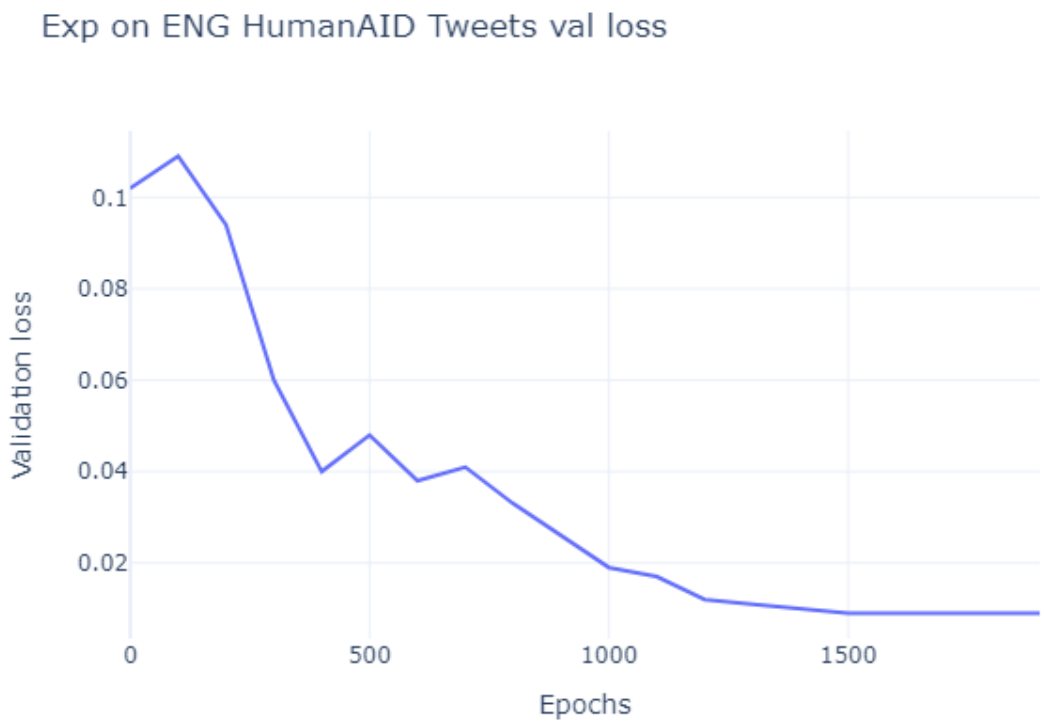

FIG. 4.4 : Évaluation de la convergence de la clustering loss lors de la validation dans l'expérimentation 1

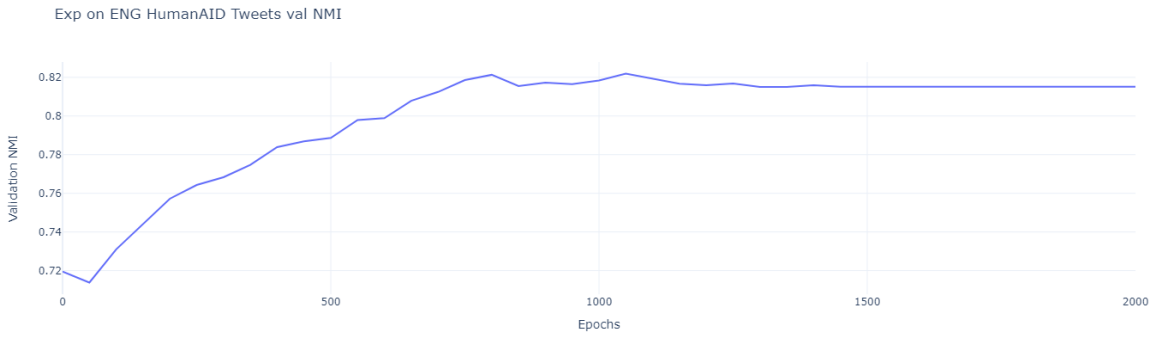

FIG. 4.5 : Évaluation de du NMI lors de la validation dans l'expérimentation 1

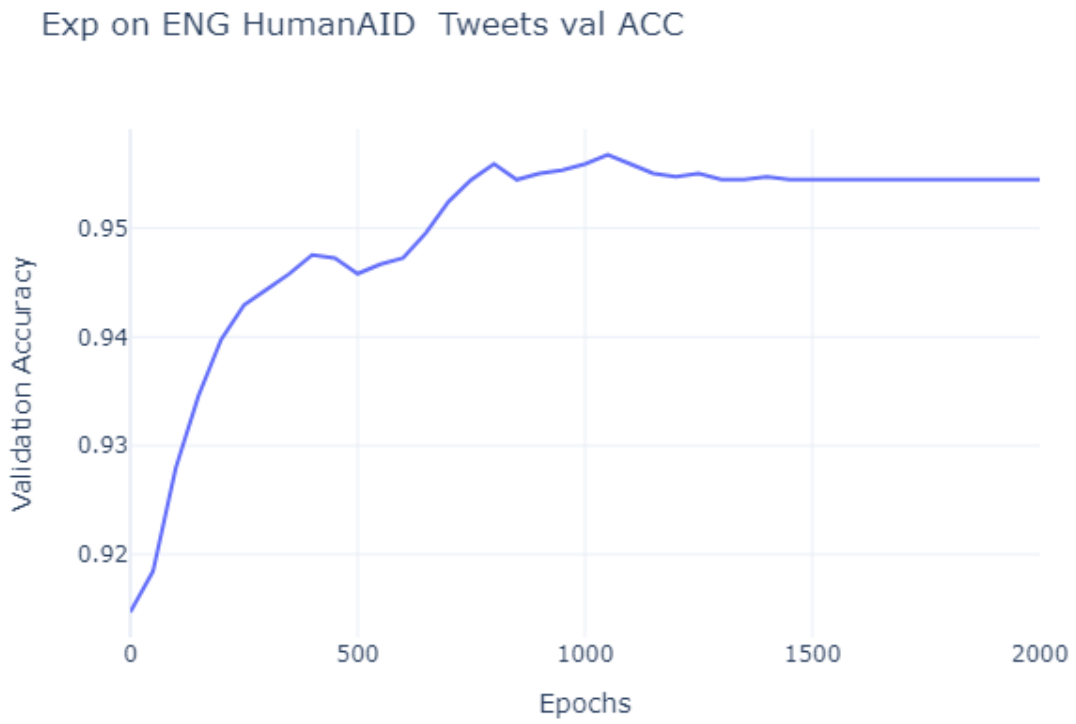

FIG. 4.6 : Évaluation du clustering accuracy lors de la validation dans l'expérimentation 1

Globalement, les résultats obtenus lors de cette expérimentation montrent que le modèle fonctionne bien sur un corpus des catastrophes naturelles en anglais,il a réussi a construire 3 cluster qui regroupe la majorité des tweets qui parlant de la même crise, autrement, le modèle regroupe les tweets similaire au niveau du contexte lié à chaque crise .

3.2 Expérimentation 2

Dans cette expérimentation, nous avons entraîné le modèle sur les données RéSoCIO (voir la section 2). Ces données concernent des tweets en francais relatifs des catastrophes naturelles de types inondations et séismes qui sont des données réelles extraites de l'API de Twitter.Nous avons entrainé et validé le modèle en utilisant les configurations et les paramètres indiqués (voir tableau 4.2). Dans l'expérimentation suivante, on a vise a tester le modèle sur un corpus de tweets en Francais, puisque notre but principale est de se concentrer sur les tweets en francais dans le cadre du projet RéSoCIO, et c'est une languequi n'est pas testé dans les travaux dans la publications du SCCL [23], ainsi que l'encodeur comme utilisé dans la première expérimentation c'est BERT et qui est conçu pour l'anglais, et donc on a décidé de changer l'encodeur, qui génère des représentations du texte, par un encodeur pré-entrainé sur le francais, et on a donc choisi CamemBERT [13].

Paramètre	Valeur
Taille du batche	400
Nb des classes	12
Nb Itération	2000
Marge CL	0.5
Taux d'apprentissage	1e-05
Encodeur	CamemBERT
Algorithme de clustering	KMeans
Temps d'exécution	5-6 hr

Tab. 4.2 : Configuration du SCCL de la deuxième expérimentation

Les résultats de cette expérimentation démontrent de mauvaises performances du modèle sur les données de validation. Ces performances peuvent être observées dans les graphiques ci-dessous. Tout d'abord, nous remarquons la divergence et la fonction de *clustering loss* qui ne se minimise pas dans le graphique 4.7. De plus, l'expérimentation donne un faible NMI, avec une moyenne inférieur à 0,15, comme évalué lors des itérations dans le graphique 4.8. On peut aussi observer la dimunition de la précision du clustering dans le graphique 4.9, avec une moyenne de faible inférieur à 0,2.

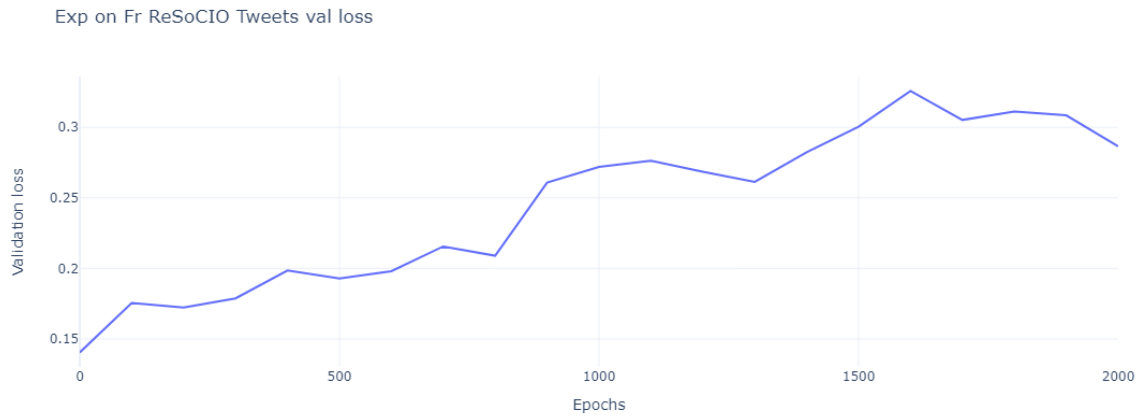

FIG. 4.7 : Evaluation de la convergence du clustering loss lors de la validation dans l'expérimentation 2

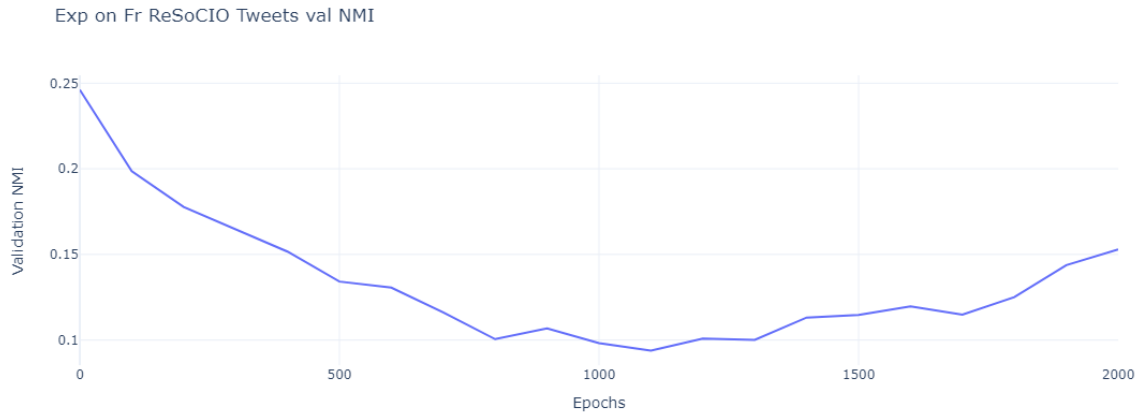

FIG. 4.8 : Evaluation de du NMI lors de la validation dans l'expérimentation 2

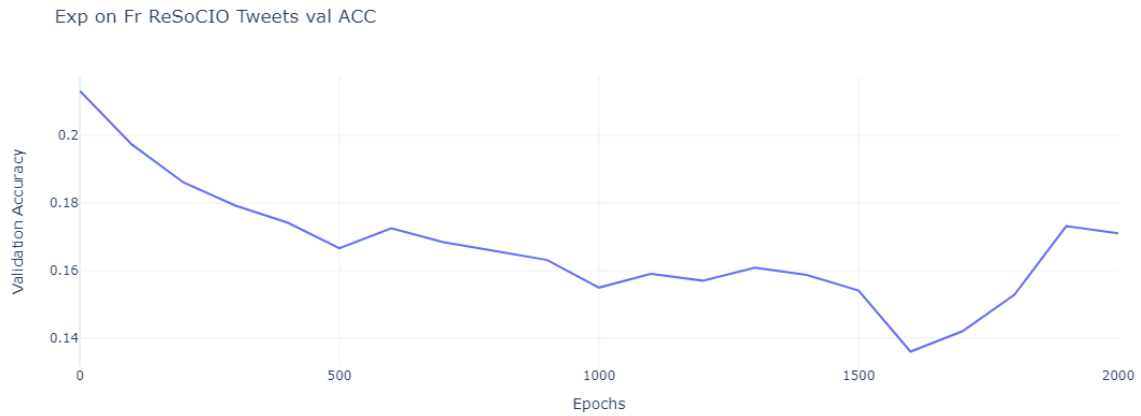

FIG. 4.9 : Evaluation du clustering accuracy lors de la validation dans l'expérimentation 2

En résumé, les résultats de cette expérimentation montrent que le modèle n'a pas réussi à apprendre les patterns nécessaires pour associer les tweets de chaque crise à un cluster. L'objectif de cette expérimentation était de tester le fonctionnement du modèle en langue française, mais malheureusement, les résultats n'ont pas été positifs. On note une différence entre les données HumanAID, sans tenir compte de la langue, et les données RéSoCIo, comme nous l'avons déjà mentionné dans les deux sections suivantes 3 et 2. Cette différence concerne la qualité des données : les données RéSoCIO contiennent du bruit et des tweets qui ne sont pas pertinents, contrairement aux données HumanAID, où les instances ne contiennent que des termes pertinents liés à chaque crise. De plus, ces données sont labellisées par des ens experts, ce qui les rend bonnes en termes de qualité. De plus, les requêtes utilisées pour extraire les tweets ne garantissent pas que ces derniers contiennent des informations pertinentes sur la crise. Par conséquent, nous pourrions par exemple trouver des poèmes ou des citations qui contiennent ces mots-clés, mais qui ne parlent pas d'une crise. Et donc, nous pouvons donc formuler l'hypothèse suivante : la langue des corpus de données n'est pas la cause du mauvais fonctionnement du modèle, mais plutôt la qualité des données RéSoCIO qui contiennent des instances non pertinentes liées à chaque crise.

Dans ce contexte, pour vérifier cette hypothèse, nous avons dû rechercher des données en français qui sont filtrées 3, comme c'est le cas pour l'expérimentation suivante que nous allons présenter.

3.3 Expérimentation 3

Dans cette expérimentation, nous avons entraîné le modèle sur les données filtrés (voir la section 3). Ces données concernent des tweets en francais relatifs des catastrophes naturelles qui sont des données réelles extrait de l'API de twitter, mais ils sont annotés par l'humain en se basant sur leur contenu. Nous avons entraîné et validé le modèle en utilisant les configurations et les paramètres indiqués (voir tableau 4.3). Dans l'expérimentation suivante, Comme dans la deuxième expérimentation on a changé l'encodeur par CamemBERT [13] .

Paramètre	Valeur
Taille du batche	400
Nb des classes	12
Nb Itération	2000
Marge CL	0.5
Taux d'apprentissage	1e-05
Encodeur	CamemBERT
Algorithme de clustering	KMeans
Temps d'exécution	5-6 hr

Tab. 4.3 : Configuration du SCCL de la troisième expérimentation

Les résultats de cette expérimentation démontrent aussi de mauvaises performances du modèle sur les données de validation. Ces performances peuvent être observées dans les

graphiques ci-dessous. Tout d'abord, nous remarquons la divergence et la fonction de *clustering loss* qui ne se minimise pas dans le graphique 4.10. De plus, l'expérimentation donne un faible NMI, avec une moyenne inférieur à 0,25, comme évalué lors des itérations dans le graphique 4.11. On peut aussi observer la diminution de la précision du clustering dans le graphique 4.12, avec une moyenne de faible inférieur à 0,32. tous ces résultats, qui sont mauvais, mais on remarque qu'il y a une augmentation de 0,1 par rapport à l'expérimentation précédente avec des données non filtrés.

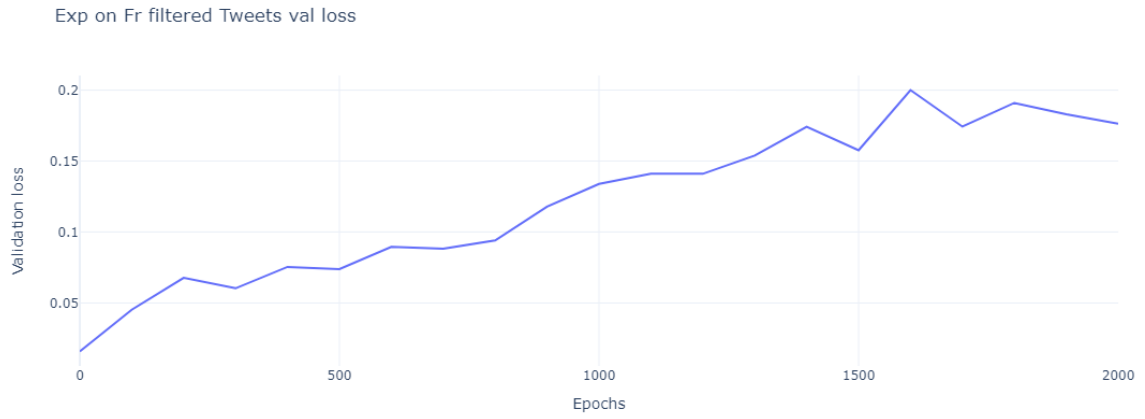

Fig. 4.10 : Evaluation de la convergence du clustering loss lors de la validation dans l'expérimentation 3

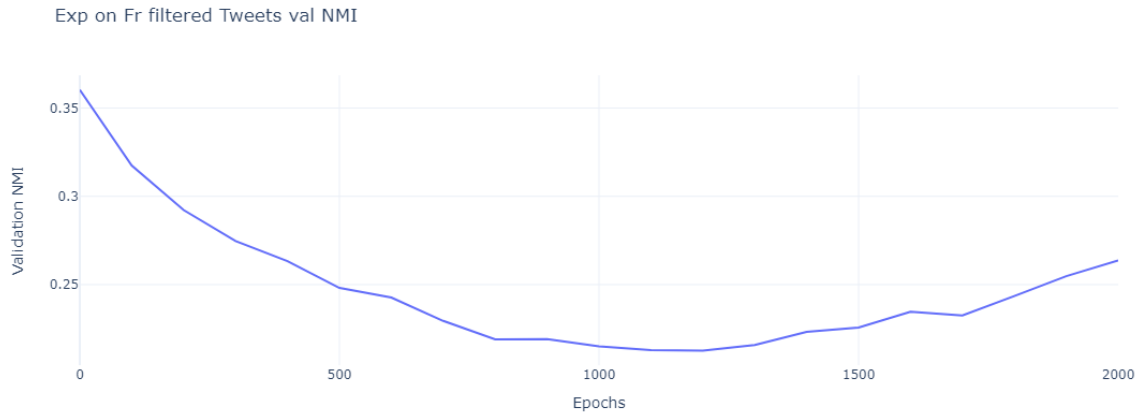

Fig. 4.11 : Evaluation de du NMI lors de la validation dans l'expérimentation 3

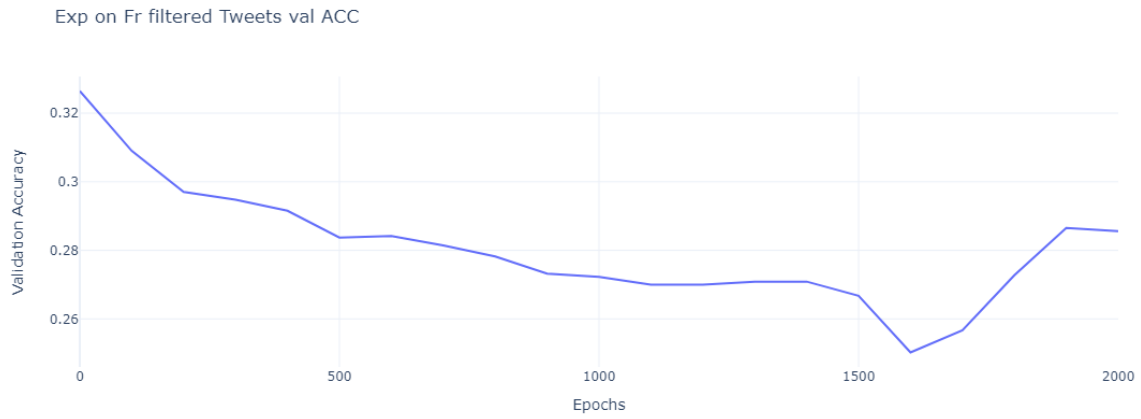

FIG. 4.12 : Evaluation du clustering accuracy lors de la validation dans l'expérimentation 3

En résumé, malgré les données filtrées contenant des termes pertinents pour chaque crise, comme nous l'avons mentionné au début dans la section 4, le modèle reste faible face aux tweets sur les catastrophes naturelles en français. Ainsi, l'idée que nous avons eue à ce moment-là était que le problème venait des représentations générées par CamemBERT. Par conséquent, nous allons essayer de pousser cet encodeur à produire des représentations similaires pour les tweets qui traitent de la même crise en utilisant une autre approche de solution.

4 Synthèse

Dans ce chapitre, nous avons présenté en détail le modèle SCCL, qui nous semble le plus approprié pour notre contexte dans la réalisation de l'état de l'art. Nous l'avons expérimenté sur trois bases de données différentes. Il a montré de bonnes performances pour la base de données en anglais, mais pour les deux en français, les résultats restent faibles, malgré la différence entre les propriétés des deux bases de données en termes de qualité et d'informations pertinentes. Après ces expérimentations, il nous est apparu nécessaire de changer d'approche. Nous sommes parvenus à la conclusion que l'encodeur pré-entraîné doit être ré-entraîné d'une autre façon pour accentuer la distinction entre les représentations des tweets au niveau du contexte relatif à chaque crise naturelle. Cette nouvelle approche sera présentée dans le chapitre suivant.

Chapitre V

Approche basé sur l'apprentissage de similarité

Après avoir étudié et expérimenté l'approche par clustering pour résoudre le problème, où l'amélioration de la similarité entre les représentations des instances par apprentissage contrastif ne constitue qu'un processus d'amélioration du processus global du clustering, nous avons conclu que l'encodeur pour le français n'est pas capable de bien suivre le processus d'apprentissage contrastif. Par conséquent, nous avons décidé de faire de la génération de représentations notre objectif principal, en nous basant sur l'apprentissage de similarité entre les différentes instances. Ainsi, dans l'approche suivante, l'objectif est de se focaliser sur le fine-tuning de l'encodeur afin d'apprendre à générer des représentations similaires pour les instances de la même catégorie.

Dans le chapitre suivant, nous présenterons le deuxième travail réalisé où nous avons opté pour l'apprentissage de similarité. Nous utiliserons le transfert d'apprentissage (Transfer Learning) pour construire un modèle capable de prédire si deux tweets parlent de la même crise ou non. Ainsi, dans ce chapitre, nous détaillerons l'architecture du modèle et explorerons les différentes expérimentations que nous réaliserons avec ce modèle.

1 Introduction

L'apprentissage de la similarité est une branche de l'apprentissage automatique qui se concentre sur l'apprentissage de la relation entre des paires d'objets, comme leur similarité ou leur dissimilarité. C'est particulièrement utile dans les scénarios où une comparaison directe des caractéristiques n'est pas pratique, ou lorsque la définition de la similarité peut dépendre du contexte ou être complexe.

Une méthode populaire d'apprentissage de la similarité consiste à utiliser des réseaux de neurones siamois [1], une architecture particulière de modèles d'apprentissage profond. Les réseaux siamois se composent de deux réseaux de neurones identiques, chacun prenant l'un des deux objets d'entrée et produisant un vecteur de caractéristiques en sortie. Ces vecteurs de caractéristiques sont ensuite comparés pour mesurer la similarité des objets d'entrée.

Dans le contexte de la similarité textuelle, les réseaux siamois peuvent aider à résoudre des tâches telles que la récupération de documents, la réponse aux questions et la détection de doublons. Par exemple, ils peuvent prendre deux phrases ou documents comme entrées et déterminer à quel point ils sont similaires en termes de contexte et de signification. Plutôt que de se contenter de regarder les caractéristiques superficielles comme les mots utilisés, les réseaux siamois peuvent capter les relations sémantiques entre les mots et le sens contextuel global du texte, fournissant une compréhension plus nuancée de la similarité.

En s'entraînant sur un grand corpus de données textuelles, les réseaux siamois peuvent apprendre une représentation riche de la langue qui capte une large gamme de similarités et de différences entre les textes. Cela peut grandement améliorer la performance des systèmes qui reposent sur la compréhension de la similarité textuelle, que ce soit pour recommander des articles similaires, trouver des réponses pertinentes à une question, ou détecter le plagiat dans les travaux écrits.

Dans notre cas, nous utiliserons ces réseaux pour les entraîner à détecter les caractéristiques sémantiques spécifiques à chaque crise. Cela nous permettra d'identifier la similarité entre les tweets dans ce contexte. Les réseaux utilisés pour cette tâche seront du type *Transformer*. Nous utiliserons spécifiquement le *Transformer* pré-entraîné CamemBERT pour l'ensemble de ce chapitre. Il sera notre encodeur principal du texte.

1.1 Mesure de la similarité

Ils existe plusieur fonction pour la mesures de similarité entre deux points, mais puisqu'on travail avec des vecteurs qui sont les représentions de chaque tweets on choisit, la fonction cosine pour la mesure de similarité. Cette dernière est une mesure qui calcule le cosinus de l'angle entre deux vecteurs. Cette mesure est très utile dans les espaces à grande dimension, comme c'est le cas avec les vecteurs de représentation de texte, où elle a tendance à donner des résultats plus fiables que d'autres mesures de distance, comme la distance euclidienne. Cette formule représente la similarité cosinus :

$$\boxed{\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \cdot \|\mathbf{B}\|}} \tag{V.1}$$

où :

- $\cos(\theta)$ est la mesure de similarité cosinus entre les deux vecteurs.
- $\mathbf{A} \cdot \mathbf{B}$ est le produit scalaire des vecteurs **A** et **B**.
- $||\mathbf{A}||$ est la norme du vecteur **A** (essentiellement, la longueur du vecteur **A**).
- $||\mathbf{B}||$ est la norme du vecteur **B** (essentiellement, la longueur du vecteur **B**).
- θ est l'angle entre les vecteurs **A** et **B**.

Lorsque la formule V.1 est utilisée pour calculer la similarité cosinus, elle donne une valeur entre -1 et 1, qui est le cosinus de l'angle entre les deux vecteurs. Une valeur de 1 signifie que les vecteurs sont identiques (l'angle entre eux est de 0 degré), une valeur de -1 signifie qu'ils sont complètement opposés (l'angle entre eux est de 180 degrés), et une valeur de 0 signifie qu'ils sont orthogonaux ou non corrélés (l'angle entre eux est de 90 degrés).

La similarité cosinus n'est pas réellement une "distance" en soi, mais plutôt une mesure de l'orientation des deux vecteurs. Autrement dit, elle ne prend pas en compte la magnitude des vecteurs, mais seulement leur direction. C'est un choix idéal pour les vecteurs de texte, car souvent la longueur d'un document (et par conséquent la magnitude de son vecteur) n'a pas beaucoup de pertinence pour la tâche de mesure de similarité, alors que l'orientation du vecteur (qui reflète les mots qu'il contient et leur fréquence) est très pertinente.Par exemple, deux tweets pourraient contenir essentiellement les mêmes mots, mais l'un pourrait être beaucoup plus long parce qu'il contient beaucoup de répétitions. Si nous utilisions une mesure de distance qui tient compte de la magnitude, comme la distance euclidienne, ces deux tweets pourraient être considérés comme très différents. En revanche, avec la similarité cosinus, ils seraient considérés comme très similaires, car leurs vecteurs pointent dans la même direction.Donc, en résumé, la similarité cosinus est un choix préférable pour mesurer la similarité entre les vecteurs de texte parce qu'elle est moins sensible à la longueur des documents et plus sensible à leur contenu.

1.2 Préparation des données

Les données utilisées dans la suite de notre chapitre et dans toutes les expérimentations sont réparties en deux ensembles : l'ensemble des paires négatives et l'ensemble des paires positives. Ces données ont été générées à partir des tweets filtrés en français (voir la section 3). Étant donné que le modèle doit apprendre les caractéristiques de chaque crise, il est important d'avoir des données filtrées qui contiennent des informations pertinentes sur chaque crise.

Nous prenons 25 tweets par crise pour l'entraînement, 5 pour la validation et 5 pour les tests finaux. À partir de ces données, nous générons des paires négatives et positives en combinant deux tweets appartenant à la même crise, en les étiquetant respectivement par 1 pour les paires positives et 0 pour les paires négatives, afin d'assurer l'équilibrage des données. Pour chaque tweet d'une crise, nous avons également pris 13 tweets des autres crises et les avons étiquetés par 0. Ainsi, pour les données d'entraînement, nous

garantissons un équilibrage des données avec 3600 paires positives et 3575 paires négatives. Pour la validation, nous avons 165 paires négatives et 120 paires positives.

Cependant, pour le test final visant à mesurer les performances du modèle dans un cas réel, nous avons laissé la construction aléatoire des paires sans équilibrage, en conservant 120 paires positives et 615 paires négatives.

2 Architecture du modèle

Dans cette partie, nous allons détailler les composants de notre architecture (voir la figure 5.1) pour la prédiction de similarité. Le modèle se compose de deux grandes parties. La première partie est responsable de la génération des représentations des paires de tweets, tandis que l'autre partie s'occupe de la prédiction de similarité.

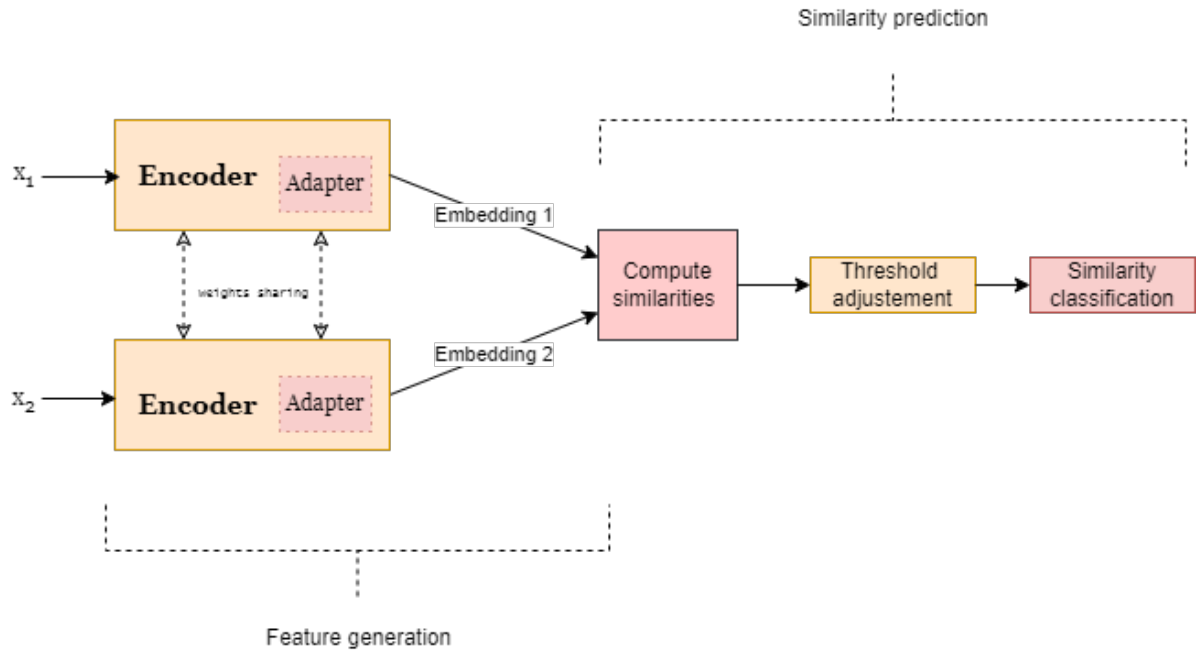

Fig. 5.1 : Architecture de notre modèle de prédiction des similarités

Le premier composant, appelé le **"Générateur des représentations"**, est l'endroit où nous avons implémenté notre réseau de neurones siamois. L'objectif de ce composant, après avoir été entraîné sur des paires de tweets positives et négatives, est de déterminer les caractéristiques communes entre les tweets relatifs à la même crise. Ainsi, il peut générer des vecteurs similaires pour les tweets liés à la même crise. Pour éviter le surapprentissage, nous avons utilisé une technique bien connue en apprentissage par transfert, qui a été publiée pour la première fois en 2019 dans l'article intitulé "Parameter Efficient Transfer learning for NLP" [1]. Cette technique consiste à utiliser des adaptateurs (voir la section 2.2) pendant l'entraînement. Les paramètres de l'encodeur principal ne sont pas modifiés, seuls les paramètres de l'adaptateur sont ajustés. Cela permet de préserver les connaissances globales acquises par l'encodeur pré-entraîné et d'exploiter les connaissances apprises par l'adaptateur pour s'adapter à la nouvelle tâche. Dans les prochaines parties,

nous allons expérimenter le modèle avec différentes configurations d'adaptateurs. Le modèle est entraîné avec un objectif contrastif, où nous utilisons la fonction du *contrastive loss* avec différentes valeurs d'objectif, afin d'encourager le modèle à produire des distances proches pour les paires positives et des distances éloignées pour les paires négatives.

Le deuxième composant, la **"Mesure de similarité"**, calcule d'abord les similarités des paires en utilisant la fonction de similarité cosinus, qui est la plus adaptée pour comparer des vecteurs. Une fois les similarités calculées, nous avons besoin d'un seuil pour classer les paires en positives et négatives. Pour cela, nous utilisons un seuil dynamique que nous ajustons à l'aide du test statistique de Kolmogorov-Smirnov (KS). Ce test est utilisé dans notre modèle pour comparer les distributions empiriques de deux ensembles de similarités : les similarités des paires positives et des paires négatives. Nous initialisons le modèle avec un seuil aléatoire pour commencer l'ajustement. Le test de Kolmogorov-Smirnov nous permet de déterminer si ces ensembles diffèrent de manière significative et nous fournit ainsi un seuil optimal pour prédire la similarité des paires.

3 Expérimentations

Dans cette partie, nous présenterons différentes expérimentations sur le modèle. Nous gardons les paramètres d'entraînement tels qu'indiqué dans le tableau 5.1. Pour chaque expérimentation, nous présenterons la configuration de l'adaptateur que nous utiliserons. De plus, pour chaque configuration, nous modifierons la marge du *contrastive loss* afin d'observer le comportement du modèle.

Paramètre	Valeur
Taille du batche	400
Nb des classes	12
Nb Itération	100
Marge CL	[0.5,0.6,0.7,0.8]
Taux d'apprentissage	2e-05
Encodeur	CamemBERT
Temps d'exécution	4 hr

Tab. 5.1 : Configuration d'entraînement du modèle

3.1 Expérimentation avec les adaptateurs de type "Bottleneck Adapters"

Les adaptateurs offrent la possibilité d'apprendre spécifiquement une tâche sans nécessiter de modifier les poids du modèle original. Ils sont insérés dans le modèle et seuls leurs paramètres sont ajustés lors de l'optimisation fine. Cette approche permet d'économiser des paramètres par rapport à une optimisation complète du modèle, et elle autorise l'utilisation simultanée de plusieurs adaptateurs pour un apprentissage multi-tâches.

Le Bottleneck Adapter est l'un de ces types d'adaptateurs qui réduit la dimensionnalité de la représentation des caractéristiques intermédiaires dans les couches du transformer. Il

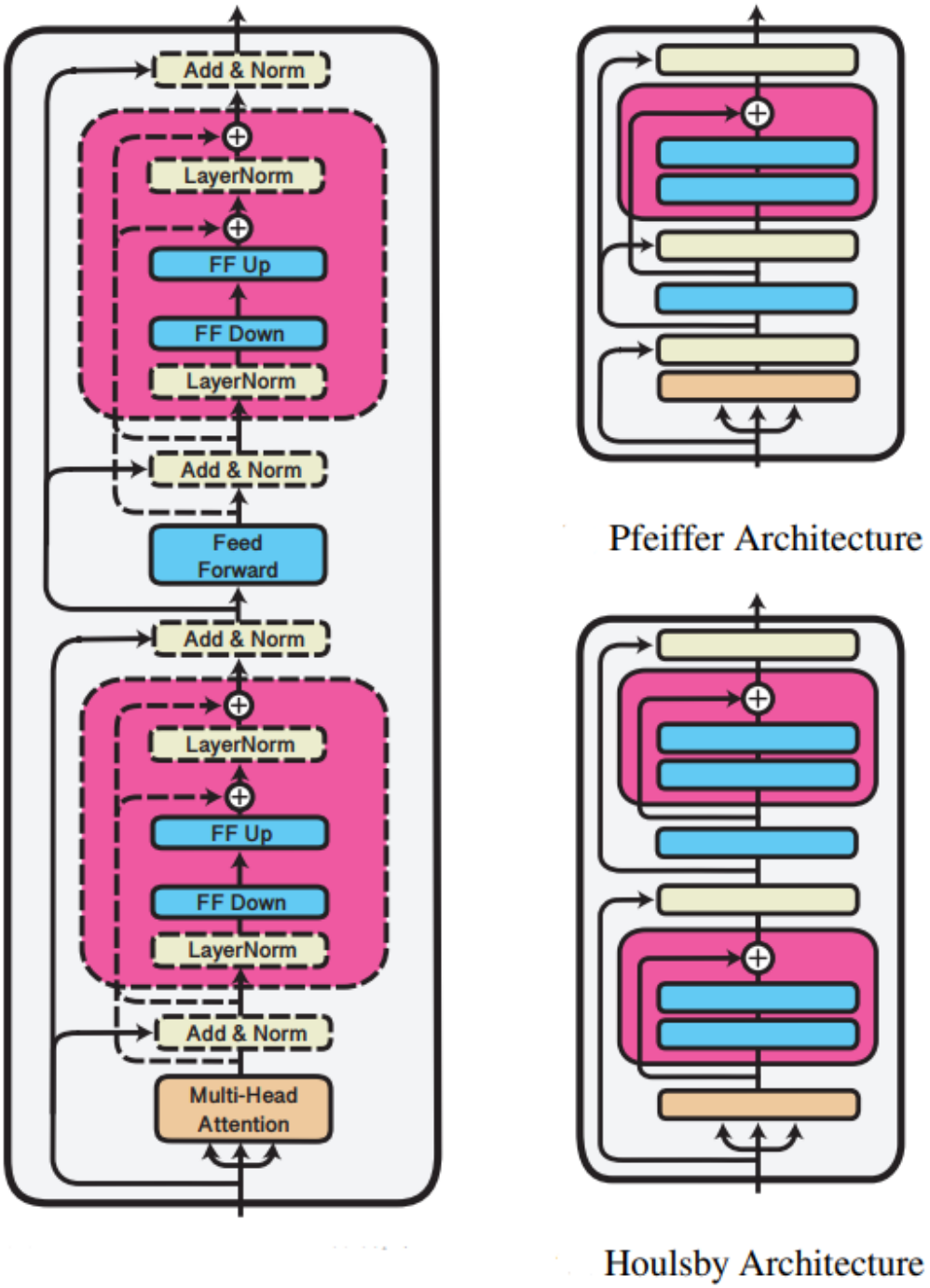

FIG. 5.2 : Architecture des configurations BottleNeck Adapters [16]

traite ensuite cette représentation réduite avant de la projeter à nouveau dans la dimension d'origine.

On trouve deux architectures populaires pour les Bottleneck Adapters : les configurations Houlsby et Pfeiffer.

Adapter Houlsby : Dans cette configuration, l'adaptateur se compose de deux couches linéaires (projection vers le bas et projection vers le haut) séparées par une non-linéarité. Cette structure en "bottleneck" réduit la dimensionnalité des activations intermédiaires au sein de la couche du transformer, ce qui réduit considérablement le nombre de paramètres à apprendre pour chaque tâche. Par exemple, si vous avez une entrée de dimension 768 et utilisez un bottleneck de dimension 64, cela vous donne environ 50 000 paramètres par adaptateur. C'est beaucoup moins que les dizaines de millions de paramètres dans un modèle transformer typique, ce qui rend cette configuration beaucoup plus efficace en termes de paramètres.

Résultats avec la configuration de Houlsby

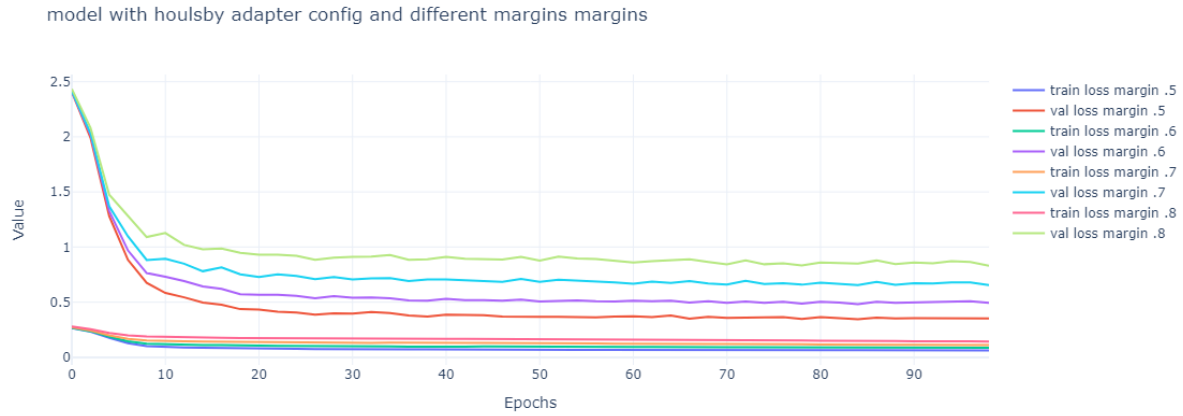

FIG. 5.3 : la fonction du *contrastive loss* pendant l'entrainement du modèle avec la configuration houlsby

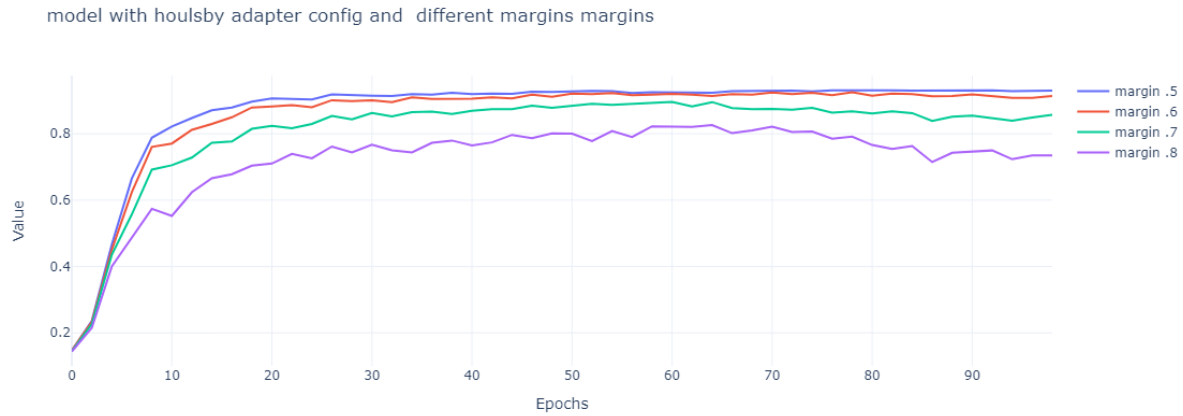

FIG. 5.4 : l'accuracy de validation pendant l'entrainement du modèle avec la configuration houlsby

Pendant l'entraînement du modèle avec la configuration Houlsby, on observe sur la figure 5.3 que la fonction de coût se minimise à chaque épisode pour l'entraînement et la

validation. Ainsi, en examinant la figure, on remarque une convergence plus satisfaisante lorsque la marge est fixée à 0,5. De plus, la figure 5.4 montre que la précision du modèle se stabilise avec de meilleures performances lorsque la marge est de 0,5. Par conséquent, le modèle présente de bons résultats de généralisation sur les données de validation.

Adapter Pfeiffer : Cette configuration étend l'adaptateur Houlsby en ayant deux adaptateurs distincts à chaque couche : un après le mécanisme d'auto-attention et un après le réseau à propagation avant. Chacun de ces adaptateurs est ensuite divisé en un adaptateur de tâche et un adaptateur de langue. L'adaptateur de tâche apprend des représentations spécifiques à une tâche, tandis que l'adaptateur de langue apprend des représentations partagées entre les tâches. Cela permet une plus grande flexibilité dans l'apprentissage de différents types de représentations pour différentes tâches, et peut potentiellement améliorer les performances dans les situations d'apprentissage multi-tâches.

Résultats avec la configuration de Pfeiffer

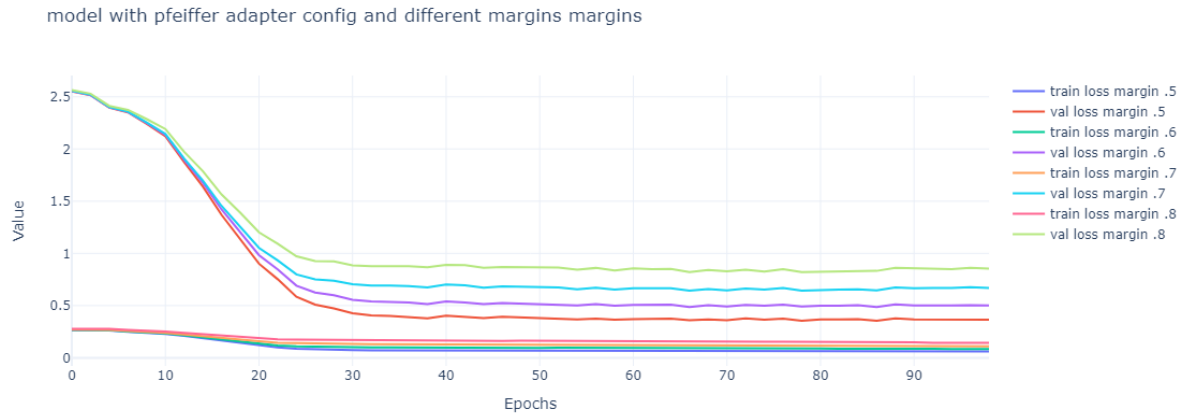

FIG. 5.5 : la fonction du *contrastive loss* pendant l'entrainement du modèle avec la configuration pfeiffer

Lors de l'entraînement du modèle avec la configuration Pfeiffer, nous constatons sur la figure 5.5 que la fonction de coût se minimise à chaque épisode pour l'entraînement et la validation. Bien qu'il y ait une légère différence par rapport à la configuration Houlsby, cette configuration montre également de bonnes performances en termes de validation pour une marge de 0,5, comme le montre la figure 5.6.

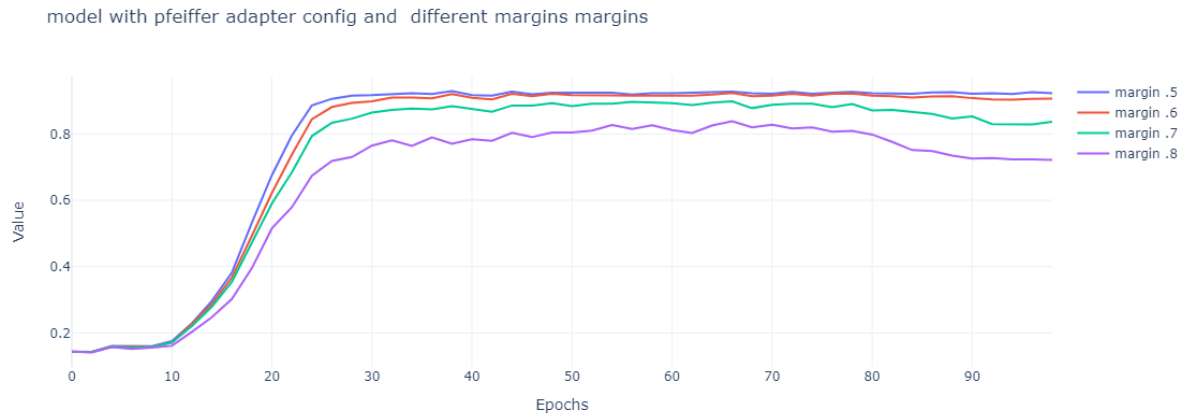

FIG. 5.6 : l'accuracy de validation pendant l'entrainement du modèle avec la configuration pfeiffer

3.2 Expérimentation avec l'adaptateur de type "Compacter"

La configuration d'adaptateur COMPACTER proposée dans la publication "COMPACTER : Efficient Low-Rank Hypercomplex Adapter Layers" introduit les matrices de poids spécifiques à chaque tâche sont insérées dans les poids du modèle préentraîné, et elles sont calculées efficacement comme une somme de produits de Kronecker entre les poids lents partagés et les matrices rapides de rang un. Cette configuration réduit le nombre de paramètres à affiner pour chaque tâche, offrant ainsi un meilleur équilibre entre les performances des tâches et le nombre de paramètres entraînables par rapport aux méthodes précédentes.

le COMPACTER utilise les trois concepts clés suivants :

- Produit de Kronecker : Il s'agit d'une opération mathématique sur deux matrices qui donne une nouvelle matrice plus grande. Dans le contexte de COMPACTER, le produit de Kronecker est utilisé pour calculer efficacement les matrices de poids spécifiques à chaque tâche.

- Poids lents et matrices rapides de rang un : Les poids lents sont partagés entre différentes tâches et changent lentement lors de l'entraînement. Les matrices rapides de rang un, quant à elles, sont spécifiques à chaque couche COMPACTER et peuvent s'adapter rapidement à chaque tâche. Une matrice de rang un peut être représentée comme le produit externe de deux vecteurs, ce qui la rend particulièrement efficace à calculer et à stocker.

- Couches de multiplication hypercomplexe paramétrées : Les nombres hypercomplexes généralisent les nombres complexes à des dimensions supérieures. La multiplication hypercomplexe peut fournir un ensemble de transformations plus riche que la multiplication standard (à valeurs réelles), ce qui peut améliorer la capacité du modèle à apprendre des motifs complexes spécifiques à une tâche. Dans le contexte de COMPACTER, il est probable que des couches de multiplication hypercomplexe paramétrées soient utilisées pour effectuer des transformations sur les

données d'entrée, bien que les détails dépendent de la mise en œuvre spécifique.

Le principal avantage de cette approche réside dans la réduction du nombre de paramètres et des coûts de calcul, ce qui la rend plus efficace à entraîner et à utiliser dans la pratique. Le compromis est que le modèle pourrait ne pas être en mesure d'apprendre des représentations spécifiques à une tâche aussi flexibles ou complexes que dans une approche de fine-tuning complète ou une architecture d'adaptateur plus complexe. Cependant, l'utilisation de couches de multiplication hypercomplexe paramétrées pourrait aider à atténuer cette limitation potentielle.

Résultats avec la configuration de Compacter

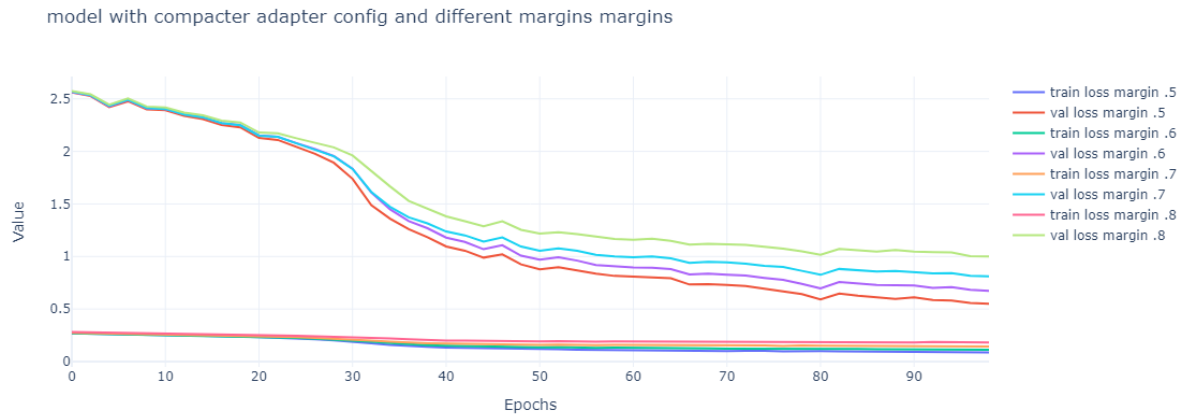

FIG. 5.7 : la fonction du *contrastive loss* pendant l'entrainement du modèle avec la configuration Compacter

FIG. 5.8 : l'accuracy de validation pendant l'entrainement du modèle avec la configuration Compacter

En entraînant le modèle, nous avons remarqué, d'après les deux figures 5.7 et 5.8, que l'apprentissage est plus lent et moins performant par rapport aux deux configurations précédentes de type "BottleNeck Adapters". Cela peut s'expliquer par la complexité

des opérations mathématiques, qui ralentit l'apprentissage, ainsi que par l'application d'une modification de poids dans chaque couche. Cela peut limiter la capacité à capturer l'information sémantique globale, ce qui est crucial pour les tâches de similarité.

3.3 Test final

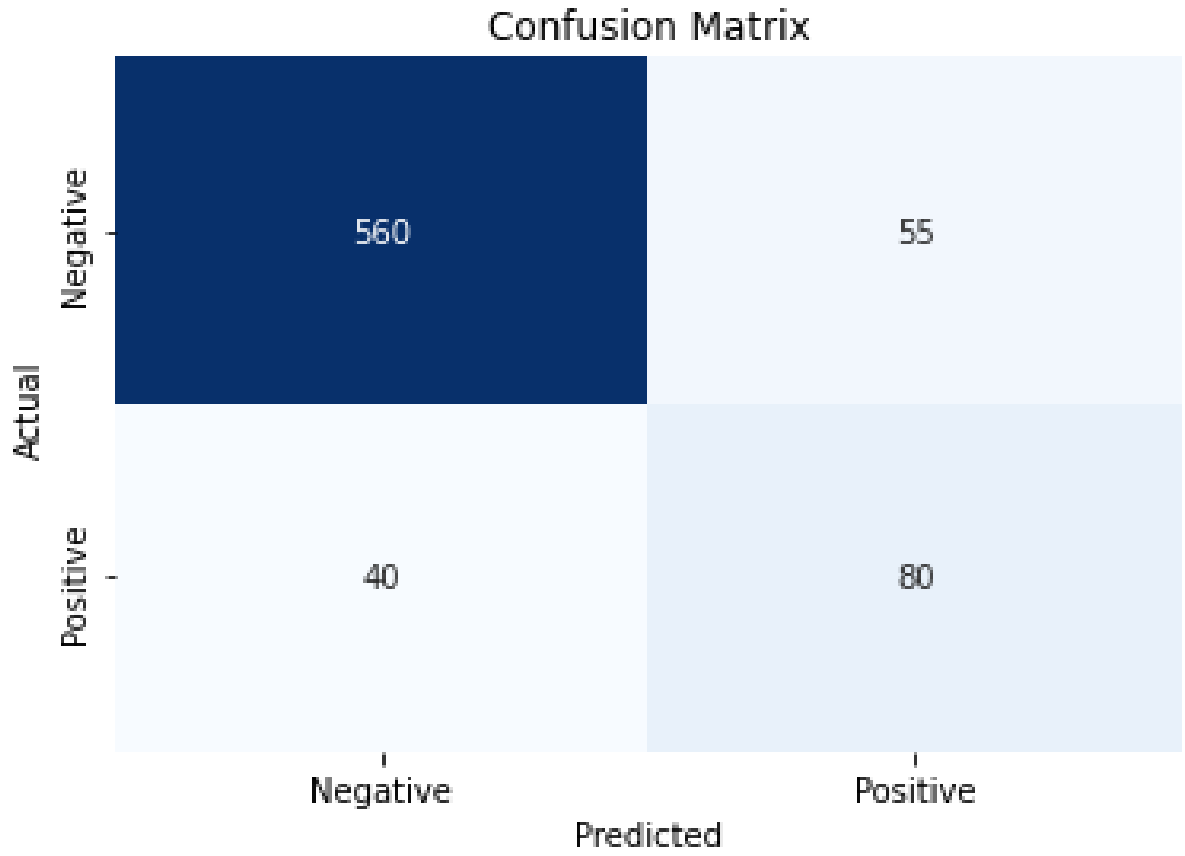

Fig. 5.9 : matrice de confusion du test final du modèle

D'après les expériences précédentes, nous avons choisi le modèle avec la configuration de Houlsby et une marge de 0.5 pourle *contrastive loss* . La performance de la configuration de Houlsby ne diffère pas significativement de celle de la configuration de Pfeiffer. Cependant, la configuration Compacter est très lente à cause de la complexité des opérations mathématiques. Quant à la marge de 0.5, elle semble être la plus stable pour toutes les expériences, suggérant que le modèle s'entraîne bien sur des paires qui sont difficiles à différencier.

En cette phase, on a décider de tester le modèle sur un autre ensemble des données de test qui sont déséquilibrées, et donc on rajoutant les métriques suivant pour mieux interprété les résultats du tests :

- **Matrice de confusion :** Une matrice de confusion est une disposition en tableau qui visualise la performance d'un algorithme. Chaque ligne de la matrice représente les instances dans une classe réelle tandis que chaque colonne représente les instances

dans une classe prédite. Les quatre résultats sont : Vrais Positifs (VP), Faux Positifs (FP), Vrais Négatifs (VN), et Faux Négatifs (FN). La matrice de confusion nous aide à comprendre les types d'erreurs que notre classificateur fait et comment il performe.

- ***Accuracy :***La précision est le ratio des observations correctement prédites sur le total des observations. C'est la mesure la plus intuitive, mais lorsque nous avons un problème de classe déséquilibrée, elle pourrait nous donner une fausse impression de la performance du modèle. En gros, la précision nous donne une idée générale de combien notre modèle prédit correctement les classifications. La formule est : $\frac{\text{VP} + \text{VN}}{\text{VP} + \text{FP} + \text{FN} + \text{VN}}$
- ***Precision :*** La precision est le ratio des observations positives correctement prédites sur le total des observations positives prédites. Si le coût d'une fausse prédiction positive est élevé, la précision est une mesure importante. Par exemple, pour la détection de spam par courrier électronique, la précision nous permet de savoir combien nos détections de spam sont correctes.

  La formule est : $\frac{\text{VP}}{\text{VP} + \text{FP}}$
- ***Recall*** Le rappel est le ratio des observations positives correctement prédites sur toutes les observations dans la classe réelle. Si le coût d'une fausse prédiction négative est élevé, le rappel est une mesure importante. Par exemple, dans la prédiction de maladies, nous préférerions un modèle avec un rappel élevé parce que manquer des cas positifs réels peut être fatal.

  La formule est : $\frac{\text{VP}}{\text{VP} + \text{FN}}$
- ***F1-Score*** Le score F1 est la moyenne pondérée de la précision et du rappel. Il tente de trouver un équilibre entre la précision et le rappel. Il est particulièrement utile si nous avons une distribution de classe inégale. Un score F1 élevé signifie que la précision et le rappel sont tous deux élevés. Le score F1 peut nous aider à comprendre comment notre modèle se comporte globalement, en tenant compte à la fois de le *Precision :* et *Recall*. La formule est : $2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Metrique	**Valeur**
Accuracy	0.8714
Precision	0.5926
Recall	0.6667
F1 Score	0.6276

Tab. 5.2 : métriques de classification des similarités

Interprétation des résultats Pou évaluer les performances de notre modèle, on a utilisé dans un premier temps la matrice de confusion qui fiure dans la figure 5.9. A partir de cette matrice,on peut dire que le modèle a correctement identifié 80 fois que deux tweets parlaient du même événement, ce qu'on appelle des "vrais positifs". Par ailleurs, il a réussi à reconnaître 560 fois que deux tweets ne parlaient pas du même événement,

ce sont des "vrais négatifs". Cependant, le modèle s'est trompé 55 fois en pensant que deux tweets parlaient du même événement alors qu'ils ne le faisaient pas, ces erreurs sont appelées "faux positifs". De plus, il a manqué 40 occasions où deux tweets parlaient du même événement, mais il les a mal identifiées comme ne parlant pas du même événement, ce sont des "faux négatifs".

En plus , a partir des résultats dans le tableau 5.2, on peut tirer les interprétations suivantes :

Une *Accuracy* de 0.8714 indique que le modèle a réussi à classer correctement environ 87.14% des tweets, déterminant avec succès si deux tweets parlent ou non du même événement.

Une *Precision* de 0.5926 révèle que, lorsque le modèle prédit que deux tweets parlent du même événement, il est correct environ 59.26% du temps. Cela suggère un taux relativement élevé de faux positifs - les cas où les tweets sont indiqués comme parlant du même événement alors qu'ils ne le font pas.

Un *Recall* de 0.6667 signifie que le modèle a identifié correctement 66.67% des paires de tweets qui parlaient réellement du même événement. Cela signifie également un taux assez élevé de faux négatifs - les cas où les tweets sont indiqués comme ne parlant pas du même événement alors qu'ils le font en réalité.

Le *F1-score* de 0.6276 est une moyenne pondérée de la précision et du rappel. Malgré une *Accuracy* élevée de 0.8714, la *Precision* plus faible de 0.5926 et le *Recall* de 0.6667 réduisent le score F1 à 0.6276.

En somme, malgré le déséquilibre des données du test, le modèle s'en sort plutôt bien. Il réussit à déterminer correctement la majorité du temps si deux tweets parlent du même événement ou non, même s'il commet parfois des erreurs. Il semble toutefois avoir plus de difficulté à identifier correctement quand deux tweets parlent du même événement, en particulier par rapport à quand ils n'en parlent pas.

4 Synthèse

Dans ce chapitre, on a présenté la deuxième solution qu'on a expérimenté, on a présenter le modèle d'apprentissage de similarité qui a pour but de forcer l'encodeur des tweets à aprrendre de générer des *embedding* qui sont très proches au terme de la distance, on a donc choisir de faire un fine-tuning avec différent configurations d'adaptateurs, et avec un ajustement dynamique de la seuil pour mesurer et classifier la similarité contextuelle entre deux tweets.

Par la suite, on a présenté les différents expérimentation sur le modèle où on a conclut plusieurs choses. Tout d'abord, la comparaison entre les configurations de Houlsby et Pfeiffer et l'adaptateur COMPACTER a montré que les premiers offraient plus de flexibilité. Bien que COMPACTER soit performant d'un point de vue de calcul grâce à son utilisation d'une approximation de rang un et d'une optimisation de faible rang, il est très lents au terme d'apprentissage des caractéristiques relative à notre tache. Ensuite, les paramètres et stratégies d'initialisation des configurations de Houlsby et Pfeiffer présentent un avan-

tage. Les deux modèles utilisent des paramètres indépendants et entraînables pour chaque tâche, apprenant potentiellement plus efficacement les spécificités des tâches. L'approche de Pfeiffer est relativement plus simple, en ajoutant des couches qui peuvent être gérée au modèle Transformer, qui contrôle l'ensembles des connaissances originales et spécifiques à la nouvelle tâche. En revanche, l'adaptateur de Houlsby est plus sophistiqué, exploitant la non-linéarité pour comprendre les patterns complexes dans les données. En outre, L'adaptateur de Houlsby s'est révélé supérieur, grâce à ses capacités non linéaires. Ces capacités lui ont permis de comprendre et d'apprendre à partir des données de manière plus nuancée par rapport à l'adaptateur de Pfeiffer. Par ailleurs, la configuration COMPACTER a une configuration différente utilisant un poids lent partagé et une matrice de rang un rapide, ce qui pourrait limiter sa capacité à apprendre des détails spécifiques à la tâche. Bien que COMPACTER utilise des couches de multiplication hypercomplexes paramétrées pour des transformations potentiellement plus riches, cela pourrait ne pas fournir un avantage significatif à moins que la tâche ne bénéficie substantiellement de ces transformations.

En plus, le choix d'une marge de 0.5 dans la fonction du *contrastive loss* s'est avéré efficace dans ces tests. La marge de 0.5 se concentre sur l'apprentissage à partir des paires difficiles dans les données plutôt que des paires faciles, ce qui améliore le processus d'apprentissage pour les tâches avec beaucoup de paires faciles. Finalement, les expériences ont prouvé l'efficacité de l'adaptateur de Houlsby et la marge de 0.5 dans la fonction du *contrastive loss* pour améliorer le processus d'apprentissage dans les tâches de similarité.

Enfin, nous avons testé un modèle de classification sur des données déséquilibrées, évalué par plusieurs métriques dont la matrice de confusion, l'accuracy, la precision, le recall et le F1-score. Bien que le modèle ait montré une accuracy élevée, il présentait des problèmes en termes de precision et de recall. En dépit de sa capacité à classifier la majorité des tweets correctement, il a eu du mal à distinguer les cas similaires. Cela signifie qu'il existe des opportunités d'amélioration pour réduire les erreurs et optimiser la performance du modèle.

Conclusion et perspectives

Conclusion générale

Dans cette étude, nous avons examiné l'utilisation des réseaux sociaux, notamment Twitter, dans la gestion des catastrophes naturelles. Nous avons mis en évidence l'importance croissante des réseaux sociaux dans notre société et leur capacité à fournir des informations en temps réel, ce qui révolutionne notre mode de communication et peut être utilisé pour aider à gérer les crises liées aux catastrophes naturelles.

Pendant ce stage, nous avons travaillé sur le projet RéSoCIO, qui cherche à utiliser les données des réseaux sociaux pour améliorer la gestion des catastrophes naturelles. Notre objectif principal était de trouver un moyen d'identifier et de regrouper les tweets liés au même événement. Pour cela, nous avons examiné différentes approches existantes dans la littérature, en mettant en évidence l'importance de la méthode SCCL basée sur le clustering pour notre sujet.

Dans un premier temps, nous avons analysé les données collectées à partir de Twitter, en nous concentrant sur la répartition des tweets par crise ainsi que les mots fréquemment utilisés dans les tweets liés à chaque crise. Nous avons constaté que certaines données étaient riches en informations, annotées par des humains, tandis que d'autres contenaient du bruit sous la forme de tweets hors contexte, principalement dus à des requêtes qui ne garantissent pas la pertinence de la majorité des données. Cela nous a également posé des défis, notamment la nécessité de filtrer et de traiter ces tweets dans leur ensemble.

En mettant en œuvre la méthode SCCL, nous avons obtenu des résultats encourageants, en particulier pour les bases de données en anglais. Cependant, nous avons également identifié des limitations pour les bases de données en français, ce qui nous a conduit à repenser notre approche.

Nous avons donc adopté une nouvelle approche basée sur l'apprentissage de similarité entre les instances. En utilisant le transfert d'apprentissage, nous avons construit un modèle pour prédire si deux tweets parlent de la même crise ou non. Cette approche a montré de bons résultats, mais des améliorations sont encore nécessaires, notamment en ce qui concerne la gestion des données déséquilibrées.

En résumé, ce stage nous a permis de mieux comprendre le contexte général de l'utilisation des réseaux sociaux pour la gestion des catastrophes naturelles. Nous avons exploré différentes approches et mené des expérimentations avec différentes données dans le but de contribuer à l'amélioration de la connaissance situationnelle des catastrophes naturelles en détectant les tweets similaires dans le contexte des crises. Nous avons obtenu des ré-

sultats prometteurs. Bien que des défis subsistent, ce travail constitue une base exploitée et intégrée à l'écosystème du projet.

Perspectives

Les perspectives de notre sujet dans le projet RéSoCIO sont prometteuses et ouvrent la voie à de nombreuses opportunités d'amélioration. Tout d'abord, nous prévoyons d'enrichir nos données et d'élargir notre analyse en réalisant des tests sur d'autres données, notamment celles du projet RéSoCIO. Nous attendrons qu'elles soient correctement filtrées afin d'augmenter la quantité de données d'entraînement et de test, ce qui nous donnera une vision claire et globale des performances de notre modèle.

De plus, nous envisageons d'explorer des méthodes avancées d'apprentissage profond, telles que l'apprentissage sensible au coût (*Cost Sensitive Learning*), pour résoudre le problème des données déséquilibrées. Cela pourrait également inclure l'utilisation d'autres modèles de langage pré-entraînés sur le corpus français et l'architecture des adaptateurs pour une meilleure classification et regroupement des tweets.

Une autre perspective intéressante serait de simuler en temps réel le comportement de notre modèle en mettant en place une API qui reçoit un flux de tweets, comprenant à la fois des instances relatives au même événement et d'autres non. En construisant des paires à partir de ce flux, notre modèle classera les paires positives et négatives. Ensuite, en utilisant l'algorithme de l'« Union Find » (voir l'annexe 2), nous regrouperons ces paires pour obtenir finalement des groupes d'instances similaires pour chaque événement. Cela nous aidera à avoir une idée globale du comportement de notre modèle sur un flux de données.

Enfin, nous souhaitons industrialiser notre modèle en passant à la phase de déploiement et en le transformant en un modèle basé sur l'apprentissage en ligne (*Online Learning*), sous la forme d'une API qui sera intégrée dans le système de traitement des tweets en temps réel.

Dans l'ensemble, les perspectives du projet RéSoCIO sont passionnantes et offrent de nombreuses possibilités d'innovation et d'amélioration dans la gestion des catastrophes naturelles grâce à l'utilisation des réseaux sociaux.

Bibliographie

[1] Jane Bromley et al. “Signature verification using a” siamese” time delay neural network”. In : *Advances in neural information processing systems* 6 (1993).

[2] Jacob Devlin et al. “Bert : Pre-training of deep bidirectional transformers for language understanding”. In : *arXiv preprint arXiv :1810.04805* (2018).

[3] Santo Fortunato. “Community detection in graphs”. In : *Physics reports* 486.3-5 (2010), p. 75-174.

[4] John Hewitt. “Note 10 : Self-Attention & Transformers 1 1 Course Instructor”. In : ().

[5] Neil Houlsby et al. “Parameter-efficient transfer learning for NLP”. In : *International Conference on Machine Learning*. PMLR. 2019, p. 2790-2799.

[6] Wei Jin et Rohini K Srihari. “Graph-based text representation and knowledge discovery”. In : *Proceedings of the 2007 ACM symposium on Applied computing*. 2007, p. 807-811.

[7] Diksha Khurana et al. “Natural language processing : State of the art, current trends and challenges”. In : *Multimedia tools and applications* 82.3 (2023), p. 3713-3744.

[8] Sosuke Kobayashi. “Contextual augmentation : Data augmentation by words with paradigmatic relations”. In : *arXiv preprint arXiv :1805.06201* (2018).

[9] Diego Kozlowski et al. “A three-level classification of French tweets in ecological crises”. In : *Information Processing & Management* 57.5 (2020), p. 102284.

[10] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In : *Proceedings of the IEEE* 86.11 (1998), p. 2278-2324.

[11] David Liben-Nowell et Jon Kleinberg. “The link prediction problem for social networks”. In : *Proceedings of the twelfth international conference on Information and knowledge management*. 2003, p. 556-559.

[12] Yinhan Liu et al. “Roberta : A robustly optimized bert pretraining approach”. In : *arXiv preprint arXiv :1907.11692* (2019).

[13] Louis Martin et al. “CamemBERT : a tasty French language model”. In : *arXiv preprint arXiv :1911.03894* (2019).

[14] R Mervin. “An overview of question answering system”. In : *International Journal Of Research In Advance Technology In Engineering (IJRATE)* 1 (2013), p. 11-14.

[15] John X Morris et al. “Textattack : A framework for adversarial attacks, data augmentation, and adversarial training in nlp”. In : *arXiv preprint arXiv :2005.05909* (2020).

[16] Jonas Pfeiffer et al. “Adapterhub : A framework for adapting transformers”. In : *arXiv preprint arXiv :2007.07779* (2020).

[17] William J Rapaport. “Syntactic semantics : Foundations of computational natural-language understanding”. In : *Thinking computers and virtual persons*. Elsevier, 1994, p. 225-273.

[18] Shuhuai Ren et al. “Generating natural language adversarial examples through probability weighted word saliency”. In : *Proceedings of the 57th annual meeting of the association for computational linguistics*. 2019, p. 1085-1097.

[19] David E Rumelhart, Geoffrey E Hinton et Ronald J Williams. “Learning representations by back-propagating errors”. In : *nature* 323.6088 (1986), p. 533-536.

[20] Unknown. *Campaigning on Twitter : The Handbook for NGOs, Politics & Public Service*. Unknown, Unknown.

[21] Ashish Vaswani et al. “Attention is all you need”. In : *Advances in neural information processing systems* 30 (2017).

[22] Xixi Wu et al. “Clare : A semi-supervised community detection algorithm”. In : *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, p. 2059-2069.

[23] Dejiao Zhang et al. “Supporting clustering with contrastive learning”. In : *arXiv preprint arXiv :2103.12953* (2021).

Annexes

Annexe A

Technologies

1 SpaCy

Spacy est une bibliothèque Python open source spécialisée dans le traitement automatique du langage naturel. Elle offre une large gamme de fonctionnalités, telles que le traitement de texte, la tokenisation, l'étiquetage grammatical, l'analyse syntaxique et la reconnaissance d'entités nommées. Spacy est appréciée pour sa vitesse et son efficacité, ce qui en fait un choix populaire pour le TALN à grande échelle.

2 PyTorch

PyTorch est une bibliothèque open source en Python dédiée à l'apprentissage automatique (machine learning) et à la création de réseaux de neurones. Elle offre une interface flexible et dynamique pour la construction et l'entraînement de modèles d'apprentissage profond. PyTorch est appréciée pour sa simplicité d'utilisation, sa compatibilité avec le calcul sur GPU, et sa capacité à effectuer des opérations de calculs sur des tenseurs de manière efficace. Elle est également connue pour son écosystème actif, sa documentation détaillée et sa communauté engagée. PyTorch est utilisée dans de nombreux domaines de l'intelligence artificielle, y compris la vision par ordinateur, le traitement du langage naturel et la recherche en apprentissage profond.

3 HuggingFace - Transformers

Transformers est une bibliothèque open source développée par Hugging Face, spécialisée dans le traitement automatique du langage naturel basé sur les modèles de langage pré-entraînés avec les Transformers. Elle offre une interface conviviale pour utiliser, entraîner et évaluer ces modèles, qui sont largement utilisés dans des tâches telles que la génération de texte, la traduction automatique et la classification de texte. La bibliothèque est construite sur le framework PyTorch, ce qui permet une intégration fluide avec d'autres fonctionnalités d'apprentissage automatique. Elle est appréciée pour sa facilité d'utilisation, sa flexibilité et son support continu de la part de la communauté Hugging

Face.

4 AdapterHub

AdapterHub est une plateforme open source développée par Hugging Face qui permet aux utilisateurs de partager, découvrir et utiliser des adaptateurs pour les modèles de langage pré-entraînés. Les adaptateurs sont des modules légers qui peuvent être ajoutés aux modèles existants pour effectuer des tâches spécifiques de traitement du langage naturel . AdapterHub facilite l'intégration et le partage de ces adaptateurs, offrant ainsi une approche modulaire et flexible pour personnaliser les modèles de langage. La plateforme est basée sur la bibliothèque Transformers de Hugging Face et est construite sur le framework PyTorch, ce qui garantit une compatibilité et une interopérabilité optimales avec les autres outils de l'écosystème Hugging Face. AdapterHub est apprécié pour sa simplicité d'utilisation, sa communauté active et son support continu.

5 NLPaug

NLPaug est une bibliothèque open source spécialisée dans l'augmentation de données pour le traitement automatique du langage naturel. Elle offre des fonctionnalités pour générer des variations de données textuelles afin d'enrichir les ensembles de données d'entraînement. NLPaug permet de réaliser diverses transformations telles que le remplacement de mots, la suppression de mots, l'insertion de mots et bien d'autres, permettant ainsi d'augmenter la diversité et la taille des données d'entraînement. La bibliothèque est compatible avec plusieurs langages de programmation, dont Python, et peut être utilisée avec différents modèles et frameworks de TALN. NLPaug est appréciée pour sa simplicité d'utilisation, son extensibilité et son potentiel à améliorer les performances des modèles de TALN en augmentant la variété des données d'entraînement disponibles.

Annexe B

Formules et Algorithmes

1 Formules mathématiques

1.1 la divergence de Kullback-Leibler (KLD)

la divergence de Kullback-Leibler (KLD) est une mesure utilisée dans le clustering pour évaluer la différence entre les distributions de probabilité des clusters et une distribution de référence. Elle permet de quantifier la similarité entre ces distributions en utilisant une formule mathématique spécifique. La KLD est utilisée pour guider le processus de clustering en minimisant la divergence entre les distributions, afin de créer des clusters qui correspondent mieux à la distribution de référence souhaitée. Elle peut également servir de métrique d'évaluation pour mesurer la qualité des clusters obtenus. La formule mathématique de la divergence de Kullback-Leibler est la suivante :

$$\boxed{\mathrm{KLD}(p||q) = \sum p(x) \log\left(\frac{p(x)}{q(x)}\right)} \tag{B.1}$$

où $p(x)$ représente la distribution de probabilité réelle des données d'un cluster donné, et $q(x)$ représente la distribution de probabilité de référence.

1.2 Le test de Kolmogorov-Smirnov

Le test de Kolmogorov-Smirnov est un test non-paramétrique qui est utilisé pour comparer une distribution de probabilité à une autre (dans le cas du test à deux échantillons) ou à une distribution théorique (dans le cas du test à un échantillon).

Pour une distribution cumulée empirique F_n et une distribution cumulée théorique F_0, la statistique de test de Kolmogorov-Smirnov D_n est définie par :

$$\boxed{D_n = \sup_x |F_n(x) - F_0(x)|} \tag{B.2}$$

où $\sup_x$ désigne le supremum de l'ensemble sur x.

Pour le test à deux échantillons, si on a deux distributions cumulées empiriques F_{n1} et F_{n2}, alors la statistique de test est définie par :

$$D_{n1,n2} = \sup_x |F_{n1}(x) - F_{n2}(x)| \tag{B.3}$$

Détermination du seuil Pour déterminer le seuil qui maximise la p-valeur, il faut noter que la p-valeur est la probabilité d'obtenir une statistique de test aussi extrême que celle observée, sous l'hypothèse nulle. Donc, maximiser la p-valeur revient à minimiser la statistique de test.

Dans le cadre du test de Kolmogorov-Smirnov, minimiser la statistique de test revient à minimiser la différence maximale entre les deux distributions cumulées. Ainsi, le seuil optimal serait celui qui correspond au point où cette différence maximale est minimisée.

2 Algorithmes

2.1 Algorithmes génération des paires pour l'entraînement

Algorithm 1 Generate Data

Input : data, n=25
Output : pos_data, neg_data

pos_data ← empty list neg_data ← empty list classes ← sorted(list(set([d[1] for d in data])))

Function `generate_data(`*data, n*`)`
  for *i* ***in*** *range(len(data))* **do**
    neg_class ← [c for c in classes if c ≠ data[i][1]] neg_counter ← 0 **for** *j* ***in*** *range(i+1, len(data))* **do**
      if *data[i][1] == data[j][1]* **then**
        pos_data.append((data[i][0], data[j][0], 1))
      end
      else if *data[j][1]* ***in*** *neg_class and neg_counter < n//2* **then**
        neg_data.append((data[i][0], data[j][0], 0)) neg_counter ← neg_counter + 1
      end
    end
  end
  return *pos_data, neg_data*
`generate_data(`*data, n*`)`

2.2 Algorithme pour groupement des tweets à partir des paires

Dans ce pseudo-code, la fonction GrouperEvenements prend une liste de paires (xi, xj, y). Si y=1 (c'est-à-dire que les instances xi et xj parlent du même événement), nous

fusionnons leurs ensembles avec l'opération Union. Après avoir parcouru toutes les paires, nous renvoyons parent, qui est un tableau où parent[i] est le parent (ou la racine) de l'instance i. Toutes les instances qui parlent du même événement auront le même parent (ou racine).

Algorithm 2 Algorithme Union-Find pour Grouper les tweets par Événements

Result : Retourner le tableau des parents après fusion des tweets par événements

```
Fonction Initialiser(n)
 for i de 0 à n-1 do
 |  parent[i] = i  rang[i] = 0
end
Fin Fonction

Fonction Trouver(x)
 if parent[x] n'est pas égal à x then
 |  parent[x] = Trouver(parent[x])
end
return parent[x]  Fin Fonction

Fonction Union(x, y)
 racineX = Trouver(x)  racineY = Trouver(y)
if racineX n'est pas égal à racineY then
 |  if rang[racineX] < rang[racineY] then
 |  |  parent[racineX] = racineY
 |  else
 |  |  parent[racineY] = racineX  rang[racineX] += 1
 |  end
end
Fin Fonction

Fonction GrouperEvenements(paires)
 n = longueur(paires)
Initialiser(n)
foreach paire (xi, xj, y) dans paires do
 |  if y est égal à 1 then
 |  |  Union(xi, xj)
 |  end
end
return parent  Fin Fonction
```